Sensometrics: Thurstonian and Statistical Models

Christensen, Rune Haubo Bojesen; Brockhoff, Per B.

Publication date:
2012

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Sensometrics: Thurstonian and Statistical Models

Rune Haubo Bojesen Christensen

Kongens Lyngby 2012
IMM-PHD-2012-271
Summary

This thesis is concerned with the development and bridging of Thurstonian and statistical models for sensory discrimination testing as applied in the scientific discipline of sensometrics. In sensory discrimination testing sensory differences between products are detected and quantified by the use of human senses. Thurstonian models provide a stochastic model for the data-generating mechanism through a psychophysical model for the cognitive processes and in addition provides an independent measure for quantification of sensory differences.

In the interest of cost-reduction and health-initiative purposes, much attention is currently given to ingredient substitution. Food and beverage producing companies are consequently applying discrimination testing to control and monitor the sensory properties of evolving products and consumer response to product changes. Discrimination testing is as relevant as ever because it enables more informed decision making in quantifying the degree to which an ingredient substitution is successful and the degree to which the perceptual properties of the product remain unchanged from end user perspectives.

This thesis contributes to the field of sensometrics in general and sensory discrimination testing in particular in a series of papers by advancing Thurstonian models for a range of sensory discrimination protocols in addition to facilitating their application by providing software for fitting these models. The main focus is on identifying Thurstonian models for discrimination methods as versions of well-known statistical models.

The Thurstonian models for a group of discrimination methods leading to binomial responses are shown to be versions of a statistical class of models known as generalized linear models. Thurstonian models for A-not A with sureness and 2-Alternative Choice (2-AC) protocols have been identified as versions of a class of statistical models known as cumulative link models. A theme throughout the contributions has been the development of likelihood methods for computing
improved confidence intervals in a range of discrimination methods including the above mentioned methods as well as the same-different test.

A particular analysis with 2-AC data involves comparison with an identicality norm. For such tests we propose a new test statistic that improves on previously proposed methods of analysis.

In a contribution to the scientific area of computational statistics, it is described how the Laplace approximation can be implemented on a case-by-case basis for flexible estimation of nonlinear mixed effects models with normally distributed response.

The two R packages sensR and ordinal implement and support the methodological developments in the research papers. sensR is a package for sensory discrimination testing with Thurstonian models and ordinal supports analysis of ordinal data with cumulative link (mixed) models. While sensR is closely connected to the sensometrics field, the ordinal package has developed into a generic statistical package applicable to statistical problems far beyond sensometrics. A series of tutorials, user guides and reference manuals accompany these R packages.

Finally, a number of chapters provide background theory on the development and computation of Thurstonian models for a range of binomial discrimination protocols, the estimation of generalized linear mixed models, cumulative link models and cumulative link mixed models. The relation between the Wald, likelihood and score statistics is expanded upon using the shape of the (profile) likelihood function as common reference.
Denne afhandling beskæftiger sig med at udvikle og bygge bro imellem Thurstonske og statistiske modeller for sensoriske diskriminationstest der anvendes i det videnskabelige fagområde kaldet sensometri. Med sensoriske diskriminationstest forsøges det at detektere sensoriske forskelle mellem produkter ved brug af de menneskelige sanser. Thurstonske modeller bidrager i denne sammenhæng med en stokastisk model for den datagenererende mekanisme via en psykofysisisk model for de kognitive processer samt et uafhængigt mål for kvantificeringen af sensorisk forskel.

I forsøget på at reducere produktionsomkostninger og at udvikle sundere produkter er der for tiden en del fokus på ingrediensudskiftning. Fødevareproducenter anvender således diskriminationstest til at kontrollere og overvåge de sensoriske egenskaber af produkter under udvikling samt forbrugerens reaktion på disse ændringer. Sensoriske Diskrimationstest er mere relevante og aktuelle end nogenlunde idet de muliggør mere velorienterede beslutninger ved at kvantificere i hvor høj grad en ingredienssubstitution er vellykket og i hvor høj grad de perceptuelle egenskaber ved et produkt vedbliver uændrede fra slutkundens perspektiv.

Denne afhandling bidrager til sensometri og sensorisk diskriminationstest ved gennem en serie artikler at udvikle Thurstonske modeller for en række populære sensoriske diskriminationstest. Afhandlingen bidrager også til deres praktiske anvendelse og udbredelse ved at tilvejebringe statistiske programpakker der kan fitte disse modeller. Et hovedtema har været at identificere Thurstonske modeller for diskriminationstest som versioner af velkendte statistiske modeller.

Thurstonske modeller for en gruppe af diskriminationstest der leder til binomielle observationer er således vist at kunne beskrives som specifikke versioner af en klasse af modeller kaldet generaliserede lineære modeller. Thurstonske modeller for ’A-not A with sureness’ og ’2 Alternative Choice’ (2-AC) testprotokoller er desuden identificeret som versioner af en klasse af modeller kaldet
'cumulative link models’. Et gennemgående tema i disse artikler har været udviklingen af likelihoodbaserede metoder for beregning af forbedrede konfidenceintervaller for en række diskriminationsmetoder inkluderende ovennævnte test foruden 'same-different’ testet.

En bestemt analyse af 2-AC data involverer sammenligning med en såkaldt 'identicality norm’. For sådanne test foreslår vi en ny teststørrelse der forbedrer tidligere foreslåede analysemetoder.

It et bidrag henvendt til fagområdet 'computational statistics’ beskriver vi hvordan Laplace approksimationen kan implementeres specifikt til enhver given applikation for fleksibel estimation af ikke-lineære mixede modeller med normalfordelte observationer.

I R pakkerne, sensR og ordinal, er de metodiske landvindinger implementeret, der er beskrevet og publiceret i videnskabelige artikler. sensR bidrager med software til analyse af sensoriske diskriminationstest via Thurstonse modeller og ordinal tilvejebringer software til analyse af ordinale data med 'cumulative link models'. Hvor sensR i udpræget grad er relevant for sensometriske applikationer, har ordinal udviklet sig til en generisk statistisk pakke der er anvendelig i problemstillinger langt udover sensometrien. En række tutorials, brugervejledninger og referencemanualer ledsager disse R pakker.

This thesis was prepared at the Technical University of Denmark (DTU), Department of Informatics and Mathematical Modelling (DTU Informatics), Mathematical Statistics Section in partial fulfillment of the requirements for acquiring the Ph.D. degree in Mathematical Statistics.

The thesis deals with Thurstonian and statistical models in sensometrics. Sensometrics is the scientific area that applies mathematical and statistical methods to problems from sensory and consumer science. The main focus is on developing statistical methods, models and software tools for Thurstonian based discrimination methods applied in sensory and consumer science.

The thesis consists of seven research papers, two R packages documented by their reference manuals and four complementary documents, and four technical chapters developed and written during the period 2008–2012. For the full list of works and publications associated with the Ph.D. see page vii.

Lyngby, April 2012

Rune Haubo Bojesen Christensen
This thesis is based on seven scientific methodological papers and three add-on packages for the statistical programming language R (R Development Core Team, 2011); two tutorials and two reports support these software packages. In addition, seven papers have been prepared, published or submitted in collaboration with other researchers during the PhD period, but which are not otherwise connected to the PhD project.

This thesis includes the following seven methodological research papers:


List of papers, software and other contributions


The following three R packages have been implemented:

- The *ordinal* package [www.cran.r-project.org/package=ordinal](http://www.cran.r-project.org/package=ordinal)
- The *sensR* package [www.cran.r-project.org/package=sensR](http://www.cran.r-project.org/package=sensR)
- The *binomTools* package [www.cran.r-project.org/package=binomTools](http://www.cran.r-project.org/package=binomTools)

The following three reference manuals accompany these R packages of which two are included in this thesis:


The following four tutorials or reports (so-called packages vignettes) have been written to support the R packages:


The following seven papers have been prepared, published or submitted in collaboration with other researchers during the PhD period. These papers are not part of the methodological developments included in this project and will not be further addressed:


In addition to a number of contributed talks and posters the following invited talks have been given on material included in this thesis:


Finally, seven papers have been reviewed for Food Quality and Preference, Journal of Statistical Software, and Computational Statistics and Data Analysis.
Acknowledgements

Foremost I wish to thank my supervisor, Professor Per Bruun Brockhoff for many enlightening discussions and always enthusiastic encouragements. Our meetings have always been held in a pleasant atmosphere and you have always been able to replenish my enthusiasm for the project.

To coauthors and collaborators: Graham Cleaver, Hye-Seong Lee, John Ennis, Daniel Ennis and Stig Mortensen; thank you for your contributions and kind criticism of my writings. I have learned a great deal from working with you.

To Jon Svendsen, Janne Fritt-Rasmussen, Ditte Green-Petersen and Sandra Stolzenbach; thank you for pleasant collaboration and for introducing me to the many real world problems of applied statistics.

Merete Hansen deserves a special thanks having shared an office and endured my company these past years. Thank you for enthusiastically reading and commenting on my writings, but foremost for the numerous conversations, the croissants and all the coffee.

Thanks are also due to current and previous colleagues at the Statistics Section, IMM, DTU for creating such a pleasant and stimulating working environment.

Above all, the most grateful of thanks goes to my wonderful wife, Rikke. Thank you for your patience and your invaluable support during this project and beyond.
Contents

Summary i
Resumé iii
Preface v
List of papers, software and other contributions vii
Acknowledgements xi

1 Introduction 1
  1.1 Overview of the Thesis 3

2 Sensory discrimination tests and Thurstonian models 11
  2.1 Introduction to sensory discrimination tests 11
  2.2 The basic Thurstonian model 12
  2.3 Psychometric functions for a selection of sensory discrimination
      protocols 14

3 Generalized linear mixed models 29
  3.1 Outline of generalized linear models 31
  3.2 Estimation of generalized linear mixed models 35
  3.3 Some issues in evaluating properties of estimators 36
  3.4 Computational methods for generalized linear mixed models 39

4 Estimation of cumulative link models with extensions 51
  4.1 Cumulative link models 51
  4.2 Matrix representation of cumulative link models 52
  4.3 A Newton algorithm for ML estimation of CLMs 53
  4.4 Gradient and Hessian for standard CLMs 54
CONTENTS

4.5 Structured thresholds ........................................... 55
4.6 Nominal effects ................................................. 56
4.7 Cumulative link models for grouped continuous observations ... 57
4.8 Likelihood for grouped continuous data ......................... 58
4.9 Implementation of CLMs in ordinal .......................... 58
4.10 Including scale effects in CLMs ............................... 60
4.11 Flexible link functions ......................................... 62
4.12 Estimation of cumulative link mixed models .................. 62

5 Likelihood ratio, score and Wald statistics 69

5.1 Score and Wald tests for a binomial parameter ............... 70
5.2 Contrasting the likelihood ratio, score and Wald statistics via the shape of the likelihood function ......................... 71
5.3 Explaining the Hauck-Donner effect via the shape of the likelihood function ............................................. 77

A Estimation and Inference in the Same-Different test 89

B Thurstonian models for sensory discrimination tests as generalized linear models 103

C Statistical and Thurstonian models for the A-not A protocol with and without sureness 115

D Estimation of the Thurstonian model for the 2-AC protocol 125

E Analysis of sensory ratings data with cumulative link models 147

F Paired preference data with a no-preference option — statistical tests for comparison with placebo data. 171

G Flexible estimation of nonlinear mixed models via the multivariate Laplace approximation 195

H Reference manual for the R package ordinal 219

I Reference manual for the R package sensR 273

J Analysis of ordinal data with cumulative link models — estimation with the R package ordinal 323

K A Tutorial on fitting Cumulative Link Models with the ordinal Package 357

L A Tutorial on fitting Cumulative Link Mixed Models with clmm2 from the ordinal Package 377
M  Statistical methodology for sensory discrimination tests and its implementation in sensR 389
Chapter 1

Introduction

This thesis deals with Thurstonian and statistical models in sensometrics. Sensometrics is the scientific area that applies mathematical and statistical methods to problems from sensory and consumer science. The main focus is on development of statistical methods, models and software tools for Thurstonian based discrimination methods applied in sensory and consumer science.

The primary aim of this project has been to unify and bridge the gap between the psychologically anchored probabilistic Thurstonian models and statistical models.

Sensory and consumer data is frequently produced and applied as the basis for decision making in the food industry worldwide. Similarly for many other industries, for instance in the car, fragrance and high end Hi-Fi/TV industries the production and interpretation of such data is also an integral part of product development and quality control. Denmark is no different — all major food industries produce and use such data. In food research, data is produced and used similar to the industrial use, and academic environments specifically for sensory and consumer sciences exist worldwide.

The development and application of statistics and data analysis in this area is called sensometrics. A common feature of sensory data is the use of human beings as measurement instruments. But humans are difficult to calibrate and people experience different perceptions from the same influences on their senses. This leads to several levels of variation and constitutes a special challenge to the sensometrician. Obtaining experimental data is often very expensive, so extracting the optimal amount of information is very important in product de-
pective aspects of individuals, several statistical methods reappear in experimental psychology and psychophysics. Medical decision making is another field in which humans assess images from MR and CT scannings, hence the same challenges apply here.

The application of novel and advanced statistical methods has been, and will most likely continue to be, conditional on the availability of software facilitating the application of the methodology. A significant part of this project has therefore focused on the development of the add-on packages ordinal and sensR for the statistical software package R (R Development Core Team, 2011).

In his seminal papers, Thurstone (1927a,b,c) developed his Law of Comparative Judgment. This “law”, has become widely acknowledged as a model for human perception of stimuli in its generality. The model describes the discrimination process and is applied in many research areas and industries to measure differences between confusable stimuli (e.g. products) or to investigate the intricacies of human perception.

The main theme of this thesis has been to bring Thurstonian and statistical models closer together. As shown in Brockhoff and Christensen (2010), Thurstonian models for some basic binomial discrimination protocols including duo-trio, triangle and m-AFC tests can be identified as versions of a class of statistical models known as generalized linear models (GLMs). This line of work was extended in Christensen et al. (2011) and in Christensen et al. (2012) in which the Thurstonian models for the A-not A with sureness protocol and the 2-AC protocol were identified as versions of so-called cumulative link models. These identifications make it possible to combine probabilistic inference with regression and ANOVA techniques for more insightful analyses, more powerful significance tests, reduced bias in parameter estimates and more accurate quantification of the statistical uncertainty.

So-called random effects versions of these models can help overcome one of the greatest challenges in sensory discrimination testing, namely the issue of replications in which multiple observations for each subject makes it possible to adjust for differences among subjects. These models also combine Thurstonian inference with regression tools, facilitate subject-specific inference and produce high powered tests of product differences. This line of work was begun in Christensen et al. (2012) for the 2-AC protocol and current work is exploring these models for other protocols.

A continuing area of interest is the use of likelihood methods for point and confidence interval estimation for problems in sensory science. Sensory discrimination protocols produce categorical data for which likelihood based confidence intervals are particular well suited and can improve much on conventional confidence intervals based on normal distribution approximations. Likelihood con-
1.1 Overview of the Thesis

Confidence intervals were shown to have many advantages over the conventional counterparts for $d'$ in the same-different protocol in Christensen and Brockhoff (2009). In later papers (referenced above) the likelihood intervals were developed for the binomial based discrimination protocols, the A-not A with (and without) sureness and the 2-AC protocol.

In support of the methodological developments I have written the open source and free software packages sensR (www.cran.r-project.org/package=sensR) and ordinal (www.cran.r-project.org/package=ordinal) for the free statistical software package, R (http://www.r-project.org/; R Development Core Team, 2011). In addition to the methodological developments, the sensR package also provides means for standard difference and similarity testing, $d'$ estimation, sample size estimation and modeling with the beta-binomial model in sensory discrimination protocols. The ordinal package has developed into a more generally applicable statistical package for ordinal scale data with complete support for the important class of cumulative link (mixed) models (CLMMs).

Ordinal scale data are commonplace in sensory and consumer science, and often analyzed with normal linear regression or ANOVA methods, or with conventional chi-square tests for contingency tables. Linear models are rarely completely satisfying because they inherently treat categorical data as continuous, and while conventional chi-square tests do treat data rightfully as categorical, they often do not lead to in-depth analyses and they do not easily extend to the replicated setting. CLMMs offer a convenient compromise in offering a regression type framework while rightfully treating data as categorical.

1.1 Overview of the Thesis

This thesis consists of a number of papers, material related to two R packages and four chapters providing technical background. Six of the seven papers included in this thesis were written for the journal of Food Quality and Preference — the main sensometrics journal. Readers of this journal will often not have a strong statistical background so the technical material has naturally been kept at a moderate level. An additional six appendices contain material associated with two R packages. In neither of these appendices has there been room for the technical and statistical background material. The main chapters of this thesis therefore contains mostly technical background material.

In the following sections, the main chapters, the journal papers, and the R packages ordinal and sensR are introduced, linked to each other and put into the appropriate context.
1.1.1 Main chapters

The main chapters of the thesis are intended to provide background and the theoretical (technical, statistical and computational) foundation for the papers and material on the R packages in the appendices.

In chapter 2 the mathematical foundation of Thurstonian models for a selection of sensory discrimination tests with binomial outcome is presented. This is background material for the use of psychometric functions for the discrimination methods in paper B and in the methodological vignette for the sensR package in appendix M.

Chapter 3 provides the technical background for estimation of GLMs and GLMMs. Though neither are directly implemented in any of the R packages, this chapter establishes the necessary foundation for the discussion of estimation of CLMs and CLMMs in chapter 4.

In almost all the contributions in this thesis the superiority of the likelihood ratio and root statistics over the Wald statistic for hypothesis tests and confidence intervals has been given some attention. However, neither in the papers written for sensometricians aimed at the journal of Food Quality and Preference, nor in the material supporting the R packages has it seemed appropriate to include technical material concerning the relations between Wald and likelihood based statistics. Score tests have been given somewhat less attention though they are, for instance, in a simple binomial setting known to have better frequentist properties than Wald and LR statistics (actual $\alpha$ is closer to nominal $\alpha$ in tests and confidence intervals) despite the three tests having the same asymptotic properties. In chapter 5 relations among the Wald, score and LR tests are considered.

The Wald and score statistics are considered as approximations to the LR statistic by approximating the (profile) log-likelihood function at the ML estimate and at the null hypothesis respectively. While from a theoretical statistics point of view no new results are presented, this chapter offers an intuitive geometrical explanation of the Wald and score statistics as approximations to the LR statistic that we haven’t found in similar detail in the literature. The Wald and score approximations to the LR statistic are worked out in one parameter settings and in multi-parameter settings with nuisance parameters. As an illustration of the insight this geometrical explanation brings, the misbehavior of the Wald statistic that can occur in logistic regression known as the Hauck-Donner effect (Hauck Jr. and Donner, 1977) is explained. It is also illustrated that the score test does not suffer from the same deficiency as the Wald statistic, but that the LR statistic is more powerful than both Wald and score tests.

1.1.2 Journal Papers

The first paper included in appendix A is written for and published in Food Quality and Preference. In this paper the same-different method is considered
1.1 Overview of the Thesis

and the maximum likelihood estimates of the model parameters derived. It was also shown how likelihood methods could provide insight using profile likelihood intervals and how information from different experiments could be combined in a likelihood based analysis.

The second paper included in appendix B is in many ways an introduction to several themes in this thesis. In this paper it is shown how Thurstonian models for several sensory discrimination tests with a binomial outcome can be identified as special versions of generalized linear models in which the psychometric functions for the discrimination protocols play the roles as inverse link functions. So-called family objects for these GLMs are implemented in the sensR package (cf. appendix I) and an important part of this package.

In the third paper included in appendix C, the Thurstonian model for the A-not A with sureness protocol is identified as a cumulative link model and it is shown how using CLMs can lead to a more insightful analysis than conventional methods. CLMs readily accommodate explanatory variables and it is discussed how the inclusion of these in various ways can be interpreted in terms of the Thurstonian model. The ordinal package (appendix H) is introduced as a toolbox to fit the relevant models.

In the forth paper included in appendix D it is shown how the Thurstonian model for the 2-Alternative Choice (2-AC) protocol can be estimated with a cumulative link model. The 2-AC protocol is an extension of the well-known 2-Alternative Forced Choice (2-AFC) test in which a “no difference” response is allowed, hence the protocol leads to trinomial data. The 2-AC protocol is not studied as much as the 2-AFC protocol so this paper fills the gap of a statistical treatment of the protocol and its Thurstonian model. As in the second and third paper a regression extension of the model is proposed utilizing cumulative link models and again the sensR and ordinal packages provide software support for the presented methodology. Supplementary material shows how the examples in the paper can be executed in R; this material is also kept updated as a vignette to the sensR package available at http://cran.r-project.org/package=sensR and included in this thesis after the paper in appendix D.

The fifth paper included in appendix E is a presentation of how replicated categorical ratings data, that is, clustered ordinal data, can be analyzed with CLMMs. Ratings appear regularly in sensory and consumer science in a number of different situations including hedonic (preference) ratings, and degree-of-difference ratings.

The sixth paper included in appendix F considers statistical tests of data from a 2-AC test including a so-called identicality norm. In a recent paper, Ennis and Ennis (2011) introduced the notion of identicality norms and proposed a chi-square test for the analysis. In this paper we show that this test behaves poorly and has a much too high type I error rate if the identicality norm is not estimated from a very large sample size. We then suggest a new statistical
test that solves this problem and behaves well for sample sizes typical of recent research (Alfaro-Rodriguez et al., 2007; Chapman and Lawless, 2005; Kim et al., 2008). This new statistic has higher power than alternative statistical tests, with the additional advantage that it can be decomposed for more insightful analyses in a fashion similar to that of ANOVA F-tests.

The seventh paper included in appendix G considers the Laplace approximation for estimation of nonlinear mixed effects models with a normally distributed response. It is shown how the Laplace approximation can be implemented on a case-by-case basis with limited programming efforts. This facilitates estimation of fairly general models that are not otherwise estimable by standard statistical software or software specially designed for nonlinear mixed models. This includes models with complicated random effects structures and user defined correlation structures for the residuals. This paper is targeted at a statistical and computational audience and it has been essential for the developments of computational methods for cumulative link mixed effects models as implemented in the \texttt{ordinal} package and discussed in chapter 4.

1.1.3 The \texttt{ordinal} package and cumulative link models

The official description of the \texttt{ordinal} package (cf. the reference manual, appendix H) reads: “This package implements cumulative link (mixed) models also known as ordered regression models, proportional odds models, proportional hazards models for grouped survival times and ordered logit/probit/... models. Estimation is via maximum likelihood and mixed models are fitted with the Laplace approximation and adaptive Gauss-Hermite quadrature. Multiple random effect terms are allowed and they may be nested, crossed or partially nested/crossed. Restrictions of symmetry and equidistance can be imposed on the thresholds (cut-points). Standard model methods are available (summary, anova, drop-methods, step, confint, predict etc.) in addition to profile methods and slice methods for visualizing the likelihood function and checking convergence.”

The \texttt{ordinal} package was originally motivated by early works on the A-not A with sureness protocol which is now published in Christensen et al. 2011, (appendix C). There was no available \texttt{R} package that could estimate the bivariate unequal-variances model, which is equivalent to a cumulative probit model with scale effects. The first function, \texttt{c1ls} for cumulative link location scale model was implemented and included in the \texttt{sensR} package. Eventually it was decided that this function and its associated methods were useful beyond \texttt{sensR} and an implementation was aimed for a wider audience in the \texttt{ordinal} package. The \texttt{ordinal} package first appeared on the comprehensive \texttt{R} archive network, CRAN (http://cran.r-project.org/package=ordinal) in March 2010. Since June 2010 code development has been hosted on \texttt{R-Forge} (https://r-forge.r-project.org/projects/ordinal/) supporting Subver-
1.1 Overview of the Thesis

1.1.4 The sensR package

The official description of the sensR package (cf. the reference manual, appendix I) reads: “sensR: Thurstonian models for sensory discrimination. The package provides methods for sensory discrimination methods; duo-trio, triangle, 2-AFC, 3-AFC, A-not A, same-different and 2-AC. This enables the calculation
of d-primes, standard errors of d-primes, sample size and power computations. Methods for profile likelihood confidence intervals and plotting are included.”

The first functions for the sensR package was written by Per Bruun Brockhoff in 2007 under a different package name. Some functions, for instance the family objects for the Thurstonian GLMs, still has the same structure, but practically all code has since been rewritten or revised by Rune Haubo B Christensen.

The sensR package first appeared on CRAN in July 2008 and since November 2011 code development has been hosted on R-FORGE (https://r-forge.r-project.org/projects/sensR/).

An overview of the support for sensory discrimination methods is provided in Table 1.1. The A-not A with sureness protocol is supported implicitly by requiring the ordinal package. Most features (indicated by columns in Table 1.1) are self explanatory. Replicated refers to the situation in which trials are replicated over assessors and the analysis takes account of the variation between assessors. For the binomial protocols this is accomplished with the beta-binomial and chance-corrected beta-binomial models, and for the A-not A, the 2-AC and the A-not A with sureness protocols, it is accomplished with CLMMs using the ordinal package. Regression analysis refers to the combination of regression or ANOVA modelling techniques with the sensory discrimination methods accomplished for the binomial protocols with special purpose generalized linear models and for the 2-AC and A-not A with sureness protocols with CLMs with probit links.

Table 1.1 appears incomplete in particular for power, sample size and simulation methods. This partly reflects that more work can be done, partly that some features are not feasible to implement, e.g. sample size estimation for models with more than one parameter, and partly that some aspects have not yet been developed, e.g. extensions of the same-different method for replicated situations and regression type extensions. Power could be implemented for A-not A with and without sureness, either using simulation as it is implemented for the same-different protocol or by generation of all possible outcomes as it is implemented for the 2-AC protocol. Simulation could also be implemented for the remaining protocols.

A non-exhaustive summary of the main function provided by or related to the sensR package is provided in Table 1.2.
1.1 Overview of the Thesis

Table 1.1: Support for sensory discrimination methods available in sensR.

<table>
<thead>
<tr>
<th>Discrimination</th>
<th>$d'$ estimation</th>
<th>Difference test</th>
<th>Similarity test</th>
<th>Power</th>
<th>Sample size</th>
<th>Simulation</th>
<th>Likelihood CI</th>
<th>Replicated</th>
<th>Regression analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duo-Trio, Triangle</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2-AFC, 3-AFC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A-not A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Same-Different</td>
<td>✓</td>
<td>✓</td>
<td>✓ (✓)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2-AC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A-not A w. Sureness</td>
<td>✓</td>
<td>✓</td>
<td>✓ (✓)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1.2: Summary of most important functions provided by or related to the sensR package.

<table>
<thead>
<tr>
<th>$d'$, CI tests</th>
<th>Power &amp; Sample size</th>
<th>Transformation</th>
<th>Illustration</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>discrim</td>
<td>discrimPwr</td>
<td>rescale</td>
<td>plot</td>
<td>findcr</td>
</tr>
<tr>
<td>AnotA</td>
<td>d.primePwr</td>
<td>psyfun</td>
<td>ROC</td>
<td>clm2twoAC</td>
</tr>
<tr>
<td>samediff</td>
<td>discrimSS</td>
<td>psyinv</td>
<td>AUC</td>
<td>SDT</td>
</tr>
<tr>
<td>twoAC</td>
<td>d.primeSS</td>
<td>pc2pd</td>
<td></td>
<td>discrimR</td>
</tr>
<tr>
<td>betabin</td>
<td>twoACpwr</td>
<td>pd2pc</td>
<td></td>
<td>discrimSim</td>
</tr>
<tr>
<td>glm</td>
<td></td>
<td></td>
<td></td>
<td>samdiffSim</td>
</tr>
<tr>
<td>clm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>clmm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduction
Chapter 2

Sensory discrimination tests and Thurstonian models

In this chapter theoretical background on Thurstonian models for a range of discrimination tests is given with some focus on the computational feasibility of the psychometric functions. Familiarity with the practical application of discrimination tests and analysis with Thurstonian models is not required to read this chapter as the topic is approached from statistical and computational perspectives. After a general introduction to sensory discrimination tests in section 2.1, the statistical basis of Thurstonian models, the notion of psychometric functions and decision rules are described section 2.2 using the 2-AFC test as an example. In section 2.3 psychometric functions are worked out for a range of other common discrimination tests.

2.1 Introduction to sensory discrimination tests

Among the most well-known sensory discrimination protocols we find the basic binomial protocols: the 2-AFC (alternative forced choice), 3-AFC, duo-trio and triangle protocols. Other protocols which are slightly more complicated include the A-not A (yes-no) and same-different methods. Rating or sureness extensions of these are known as the A-not A with sureness and the degree of difference protocols. A protocol somewhere in between is the 2-AC (alternative choice) in which a “no difference” response is allowed hence this leads to a trinomial outcome.
A basic distinction is between so-called specified and unspecified methods. The m-AFC methods are specified methods since the sensory dimension or attribute has to be specified (e.g. which of these (two/three/...) samples are most sweet/salty/...). The duo-trio and triangle protocols are unspecified since no sensory attribute is disclosed. In the triangle test a subject is presented with two sample of one kind and one of another, and the subject is asked to identify the sample most different from the other two. In the duo-trio test a subject is presented with a reference sample and two test samples; one of the same kind as the reference and one of a different kind. The subject is asked to identify the test sample most similar to the reference sample.

In a series of papers Thurstone (1927a,b,c) published the so-called law of comparative judgement which is based on the assumption that the sensory or perceptual magnitude is random. This randomness may come from variations in the stimuli (e.g. a food product) or in the neural perception of the sample. What we call Thurstonian models today are based on the same idea; that the perceptual magnitude from a given stimulus has a distribution, but the term covers situations beyond the specific situation Thurstone considered in which only two samples are compared. Today, Thurstonian models form a theoretical basis for sensory discrimination protocols and models for preferential choice. In the simplest discrimination and preference test binomial data are obtained when subjects express for instance their preference for one sample over another. In other rating and ranking situations ordinal or nominal data are obtained.

Thurstone assumed that the distributions of sensory magnitude were normal distributions probably at least in part due to convenience. Today it is also conventional to assume that the distributions are normal. There has been some debate in the literature about the suitability of this assumption (Luce, 1994). There are arguments in favor of the normal distribution based on the central limit theorem (e.g. many independent neural and stimuli-specific sources of variation will make the resulting distribution appear normal), but only in rare cases is it actually possible to empirically assess the suitability of the normal assumption.

### 2.2 The basic Thurstonian model

One of the simplest discrimination test protocols is the two-alternative forced choice (2-AFC) test protocol. When used as a discrimination protocol an assessor is given one sample from each of two stimuli and asked to identify the sample with the most or least of some attributes, say, the sweetest, most salty, most sour etc. sample. The Thurstonian model for this situation is depicted in Figure 2.1 where it is assumed that we have two stimuli with the following distributions:

\[ A \sim N(\mu_A, \sigma_A^2) \quad B \sim N(\mu_B, \sigma_B^2) \]  

(2.1)
2.2 The basic Thurstonian model

Figure 2.1: General Thurstonian model for a simple binomial two-stimulus situation.

where the horizontal axis corresponds to the relevant sensory attribute in question, e.g. saltiness.

Another aspect of Thurstonian models is the decision rule. This is the rule an assessor is assumed to apply in order to produce an answer upon perception of the samples.

For the 2-AFC discrimination protocol, the decision rule is given as follows: An assessor will respond that a sample $b$ from the $B$ distribution is of larger magnitude than a sample $a$ from the $A$ distribution if $b > a$ measured on the appropriate sensory dimension. For example a subject may respond “$b$ is more salty than $a$”. Assuming the $\mu_B > \mu_A$, a subject is said to express a correct answer if the subject responds $b > a$. The probability of a correct answer, $p_c$ is therefore related to the parameters of the stimuli distributions by:

$$p_c = P(B > A).$$

With binomial data it is not possible to identify the total of four parameters of the two stimuli distributions and some assumptions are made. In the 2-AFC model it is first of all assumed that the two stimuli distributions have the same spread: $\sigma_A = \sigma_B = \sigma$. The parameter of interest is known as the Thurstonian delta — the translation or location difference between the distributions relative to their spread: $\delta = (\mu_B - \mu_A)/\sigma$ as indicated in Figure 2.1. This is a key parameter quantifying the sensory difference between the stimuli. Since the magnitudes of the means are also not identifiable we may parameterize the Thurstonian model for this protocol solely in terms of $\delta$ without loss of generality.
as:

\[ A \sim N(0, 1) \quad B \sim N(\delta, 1). \]  

(2.2)

We can now express the probability of a correct answer, \( p_c \), in terms of \( \delta \), the Thurstonian measure of sensory difference:

\[
p_c = P(A < B) = P(A - B < 0) = P\left(\frac{A - B + \delta}{\sqrt{2}} < \frac{\delta}{\sqrt{2}}\right) = P\left(Z < \frac{\delta}{\sqrt{2}}\right) = \Phi\left(\frac{\delta}{\sqrt{2}}\right)
\]

where \( Z \) is a standard normal variate, \( \Phi \) is the standard normal cumulative distribution function, \( E[A - B] = -\delta \) and \( \text{Var}[A - B] = \text{Var}[A] + \text{Var}[B] = 2 \). This function is known as the psychometric function for the 2-AFC discrimination protocol.

### 2.2.1 Beyond the 2-AFC protocol

In section 2.2 the Thurstonian model for the 2-AFC discrimination protocol was developed introducing the basic Thurstonian model. The key result was the psychometric function relating the probability of a correct answer to the Thurstonian measure of sensory difference, \( \delta \).

The Thurstonian models relating observed data to the Thurstonian \( \delta \) were developed for the same-different test, the A-not A test, the A-not A with sureness and the 2-AC test in (Christensen and Brockhoff, 2009; Brockhoff and Christensen, 2010; Christensen et al., 2011, 2012) included in appendices A, B, C, and D respectively. Neither of these test protocols lead to simple binomial data and decision parameters in addition to \( \delta \) are involved, so the Thurstonian models cannot be summarized by simple psychometric functions.

In the remainder of this chapter, however, psychometric functions for a selection of additional discrimination protocols are developed including the triangle and m-AFC protocols used in Brockhoff and Christensen (2010), cf. appendix B and in appendix M. This contributes to a firmer mathematical foundation of the Thurstonian models for these discrimination protocols.

### 2.3 Psychometric functions for a selection of sensory discrimination protocols

In this section psychometric functions for a series of sensory discrimination protocols with binomial outcome are developed. Where possible both expressions
as single integrals and multivariate normal integrals are given. Gradients are also given where relevant, as these are needed for the implementation in sensR as family objects. Examples of the computation of the psychometric functions are illustrated in R throughout.

### 2.3.1 Derivation of psychometric functions for m-AFC

#### 2.3.1.1 The psychometric function for the 3-AFC protocol

First consider the 3-AFC protocol and assume two stimuli with the following distributions:

\[
A \sim N(0, 1) \quad B \sim N(\delta, 1)
\]

In the 3-AFC protocol three samples are presented; \(a_1, a_2\) and \(b\), and a correct answer is given if the \(b\)-sample is correctly identified, which will be the case if \(a_1 < b\) and \(a_2 < b\). The probability of a correct answer is therefore given as

\[
p_c = P(A_1 < B, A_2 < B).
\]

We will use the following identity for marginal probability density functions:

\[
P(X = x) = \int_y P(X = x, Y = y) \, dy = \int_y P(X = x | Y = y) P(Y = y) \, dy
\]

to rewrite the probability of a correct answer as

\[
p_c = \int_{-\infty}^{\infty} P(A_1 < B, A_2 < B, B = z) \, dz
\]

\[
= \int_{-\infty}^{\infty} P(A_1 < B, A_2 < B | B = z) P(B = z) \, dz
\]

\[
= \int_{-\infty}^{\infty} P(A_1 < z) P(A_2 < z) P(B = z) \, dz
\]

\[
= \int_{-\infty}^{\infty} \Phi(z)^2 \phi(z - \delta) \, dz.
\]

#### 2.3.1.2 Examples in R

Suppose we choose \(\delta = 1\), we may then evaluate the 3-AFC psychometric function by first defining the integrand function (fun) and then using a general integration routine to get the result:

```R
> delta <- 1
> fun <- function(x, delta) dnorm(x - delta) * pnorm(x)^2
> integrate(f=fun, lower=-Inf, upper=Inf, delta=delta)
0.6337021 with absolute error < 1.5e-05
```
2.3.1.3 Multivariate normal expression of the 3-AFC psychometric function

Using a change of variables, the psychometric function for the 3-AFC protocol can be expressed via the bivariate normal CDF: Define the new variables

\[ U_1 = A_1 - B \]
\[ U_2 = A_2 - B, \]

then we may express the probability of a correct response as

\[ p_c = P(A_1 < B, A_2 < B) = P(U_1 < 0, U_2 < 0) \]

Now let \( f_U(u) \) denote the multivariate normal density of \( U = [U_1, U_2]^\top \) with mean \( \mathbb{E}[U] = [-\delta, -\delta]^\top \) and variance-covariance matrix

\[
\text{VCOV}[U] = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},
\]

We may now express the probability of a correct answer as

\[ p_c = \int_{-\infty}^{0} \int_{-\infty}^{0} f_U(u) \, du, \]

which is just an evaluation of the bivariate normal CDF with limits and parameters as just defined.

The variance-covariance matrix of \( U \) is obtained by defining \( U = JX \), where \( X = [A_1, A_2, B]^\top \) and \( J \) is the Jacobian defining the transformation, where

\[
J = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix},
\]

and

\[
\text{VCOV}[U] = J\text{VCOV}[X]J^\top = JJ^\top.
\]

2.3.1.4 Examples in R

We can use the `pmvnorm` function from the `mvtnorm` package (Genz et al., 2011) to evaluate the bivariate normal CDF. Suppose again we choose \( \delta = 1 \), then we may use:

```r
> library(mvtnorm)
> S <- diag(rep(2, 2))
> S[2, 1] <- S[1, 2] <- 1
> S

[,1] [,2]
[1,] 2 1
[2,] 1 2
```
2.3 Psychometric functions for a selection of sensory discrimination protocols

> pmvnorm(lower=rep(-Inf, 2), upper=rep(0, 2), mean=rep(-delta, 2),
> sigma=S)

[1] 0.633702
attr("error")
[1] 1e-15
attr("msg")
[1] "Normal Completion"

The variance-covariance matrix of $U$ is obtained with

> J <- matrix(c(1, 0, -1, 0, 1, -1), ncol=3, byrow=TRUE)
> J %*% t(J)

<table>
<thead>
<tr>
<th>[,1]</th>
<th>[,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

We could simulate the 3-AFC task in order to estimate $p_c$ for a given $\delta$. The following code simulates the 3-AFC task 10^5 times and estimates the probability that ($B > A_1$ and $B > A_2$):

> delta <- 1
> set.seed(12345) ## for reproducibility
> A1 <- rnorm(1e5)
> A2 <- rnorm(1e5)
> B <- rnorm(1e5, delta, 1)
> mean(B > A1 & B > A2)

[1] 0.63328

Observe how this approximates $f_{m-AFC}(1)$ estimated above.

2.3.1.5 The psychometric function for the m-AFC protocol

The generalization of the psychometric function to an m-AFC protocol, where $m$ is a whole number larger than one is straightforward. Here, $m - 1$ $A$-samples and a single $B$-sample are presented. The probability of a correct answer is now

$$p_c = P(A_1 < B, \ldots, A_{m-1} < B)$$

$$= \int_{-\infty}^{\infty} \Phi(z)^{m-1} \phi(z - \delta) \, dz.$$
2.3.1.6 Derivatives of the psychometric functions for the m-AFC protocols

Before we consider the general case, we begin by considering the 2-AFC protocol. First notice the following general derivatives

\[ \frac{\partial}{\partial x} \Phi(x) = \phi(x) \quad \frac{\partial}{\partial x} \phi(x) = -x \phi(x) \]

For the 2-AFC protocol we obtain

\[ f'_{2-AFC}(\delta) = \frac{\partial}{\partial \delta} f_{2-AFC}(\delta) \]

\[ = \frac{\partial}{\partial \delta} \Phi(\delta/\sqrt{2}) \]

\[ = \phi(\delta/\sqrt{2})/\sqrt{2} \]

For the m-AFC protocol we obtain

\[ f'_{m-AFC}(\delta) = \frac{\partial}{\partial \delta} f_{m-AFC}(\delta) \]

\[ = \frac{\partial}{\partial \delta} \int_{-\infty}^{\infty} \Phi(z)^{m-1} \phi(z-\delta) \, dz \]

\[ = \int_{-\infty}^{\infty} \Phi(z)^{m-1} [\frac{\partial}{\partial \delta} \phi(z-\delta)] \, dz \]

\[ = \int_{-\infty}^{\infty} \Phi(z)^{m-1}(z-\delta) \phi(z-\delta) \, dz \]

2.3.2 The psychometric function for the triangle protocol

In the triangle test a respondent receives two samples of one kind and another sample of a different kind. The task is to pick out the odd sample. Assume we have

\[ A \sim N(0, 1) \quad B \sim N(\delta, 1) \]

The respondent is assumed to compare the distances between pairs of the three samples and pick out the sample farthest from the other two as the odd sample. Thus a correct response is made if the distance among the \textit{a}-samples are less than either of the distances between the \textit{b}-sample and the \textit{a}-samples:

\[ |a_1 - a_2| < |a_1 - b| \quad \text{and} \quad |a_1 - a_2| < |a_2 - b| \]

2.3.2.1 Expression via multivariate normal distribution

We may identify four different cases in which these inequalities are satisfied:
For instance, in the first two cases \( b \) is larger than both \( a_1 \) and \( a_2 \), and \( a_1 \) and \( a_2 \) are closer to each other than \( b \) is to either of them. In the latter two cases \( b \) is smaller than both \( a_1 \) and \( a_2 \). Now observe that the second inequalities in each of the cases \( C_1 \ldots, C_4 \) are redundant, i.e., implicitly satisfied once the first and the third inequalities are satisfied. Observe also that \( P(C_1) = P(C_2) \), and similarly that \( P(C_3) = P(C_4) \). The probability of a correct response in the triangle protocol is therefore given as

\[
p_c = 2\{P(C_1) + P(C_3)\}.
\]

Now we may write the probability of a correct response as

\[
P(C_1) = P(A_1 - A_2 < 0, 2A_2 - A_1 - B < 0) = P(V_1 > 0, V_2 > 0),
\]

where the derived variables are defined as

\[
V_1 = -A_1 + A_2
\]

\[
V_2 = A_1 - 2A_2 + B
\]

with mean \( \mathbf{E}[V] = [0, \delta]^\top \) and variance-covariance matrix

\[
\text{VCOV}[V] = \text{VCOV}[JX] = J\text{VCOV}[X]J^\top = JJ_3J^\top = JJ^\top
\]

where the Jacobian defining the variable transformation is given by

\[
J = \begin{bmatrix}
-1 & 1 & 0 \\
1 & -2 & 1
\end{bmatrix}.
\]

Now \( P(C_1) \) can be expressed as an integral over a bivariate normal density:

\[
P(C_1; \delta) = \int_0^\infty \int_0^\infty f_V(z; \delta) \, dz
\]

where \( f_V \) is the bivariate normal density with mean and variance-covariance matrix as given above.
Similarly $P(C_4)$ may be written as

$$P(C_4) = P(A_2 - A_1 < 0, A_1 - 2A_2 + B < 0) = P(V_1 < 0, V_2 < 0)$$

thus

$$P(C_4; \delta) = \int_{-\infty}^{0} \int_{-\infty}^{0} f_V(z; \delta) \, dz.$$  

Although this multivariate expression of the triangle psychometric function may not be the computationally fastest, it may prove more flexible in terms of extensions of the underlying model.

### 2.3.2.2 Examples in R

Again using `pmvnorm` from the `mvtnorm` package (Genz et al., 2011), we can compute the probability of a correct answer from a given value of $\delta$:

```r
> library(mvtnorm)
> ## define parameters:
> delta <- 1
> ## Expected value of U:
> mu <- c(0, delta)
> ## Jacobian:
> J <- matrix(c(-1, 1, 0,  
>             1, -2, 1), ncol=3, byrow=TRUE)
> ## variance-covariance matrix of U:
> (S <- tcrossprod(J))
> ## compute contributions:
> p1 <- pmvnorm(lower=c(0, 0), upper=c(Inf, Inf),
>               mean=mu, sigma=S, algorithm=TVPACK())
> p2 <- pmvnorm(lower=c(-Inf, -Inf), upper=c(0, 0),
>               mean=mu, sigma=S, algorithm=TVPACK())
> ## Pc - probability of a correct answer:
> 2 * c(p1 + p2)
> [1] 0.4180467
```

The triangle psychometric function may also be simulated. First define boolean functions for the cases $C_1, \ldots, C_4$:

```r
> f1 <- function(a1, a2, b) a1 < a2 & (a2 - a1) < (b - a2)
> f2 <- function(a1, a2, b) a2 < a1 & (a1 - a2) < (b - a1)
> f3 <- function(a1, a2, b) a1 < a2 & (a2 - a1) < (a1 - b)
> f4 <- function(a1, a2, b) a2 < a1 & (a1 - a2) < (a2 - b)
```
2.3 Psychometric functions for a selection of sensory discrimination protocols

Then simulate $10^5$ times:

```r
> set.seed(12345) ## for reproducibility
> X <- replicate(1e5, {
>     a1 <- rnorm(1)
>     a2 <- rnorm(1)
>     b <- rnorm(1, mean=delta)
>     1 * c(f1(a1, a2, b), f2(a1, a2, b), f3(a1, a2, b), f4(a1, a2, b))
> })
> ## probability contributions:
> rowSums(X) / 1e5
```

```
[1] 0.18261 0.18319 0.02575 0.02579
```

```r
> sum(rowSums(X) / 1e5) ## Pc
```

```
[1] 0.41734
```

We could also use simulation to arrive at the variance-covariance matrix for $U$:

```r
> delta <- 1
> set.seed(12345) ## for reproducibility
> ## Simulate A1, A2 and B:
> A1 <- rnorm(1e5, 0, 1)
> A2 <- rnorm(1e5, 0, 1)
> B <- rnorm(1e5, delta, 1)
> ## Generate matrix with variables corresponding to the C_1 case:
> V <- cbind(-A1 + A2, A1 - 2*A2 + B)
> round(apply(V, 2, mean)) ## means
```

```
[1] 0 1
```

```r
> round(apply(V, 2, var)) ## variances
```

```
[1] 2 6
```

```r
> ## Center the variables:
> V2 <- scale(V, center=TRUE, scale=FALSE)
> ## Sample covariance matrix:
> round(t(V2) %*% V2 / 1e5)
```

```
[,1] [,2]
[1,] 2 3
[2,] 3 6
```

2.3.2.3 Univariate expression of the triangle psychometric function

Ura (1960) states, without motivation or proof, that the probability of a correct response can be expressed as

$$p_c = 2 \int_{0}^{\infty} \left\{ \Phi \left[ -z\sqrt{3} + \delta \sqrt{2/3} \right] + \Phi \left[ -z\sqrt{3} - \delta \sqrt{2/3} \right] \right\} \phi(z) \, dz,$$

(2.3)
This appears to be correct, but an even simpler expression was arrived at by Bradley (1963). He claimed (also without motivation or proof) that a correct response will be made if the following inequality is satisfied:

\[
\frac{2|b - \frac{1}{2}(a_1 + a_2)|}{\sqrt{3}|a_1 - a_2|} > \sqrt{3}
\]

(2.4)

We can develop this inequality by first letting \( \bar{a} \) denote the midpoint between \( a_1 \) and \( a_2 \). A correct response will be made if \( b \) is further apart from either of the \( a \) samples than they are from each other. Figure 2.2 shows the break even point in which \( b \) is the same distance from \( a_2 \) as \( a_2 \) is from \( a_1 \). As is clear from Figure 2.2, \( b \) is further from the \( a \) samples than they are from each other if one third of the distance between \( \bar{a} \) and \( b \) \(|b - \bar{a}|\) is larger than one half of the distance between \( a_1 \) and \( a_2 \) \(|a_1 - a_2|\). Equivalently we have that a correct answer is made if

\[
\frac{1}{3}|b - \bar{a}| > \frac{1}{2}|a_1 - a_2|
\]

Upon noting that \( \bar{a} = \frac{1}{2}(a_1 + a_2) \), equation (2.4) is obtained after straightforward manipulations.

Because the left-hand-side of (2.4) is the absolute value of a t-statistic with one degree of freedom, it may be evaluated via the non-central \( F_{1,1} \) distribution:

\[
p_c = P(F_{1,1} > 3; \lambda = 2\delta^2/3)
\]

where \( \lambda \) is the non-centrality parameter\(^1\).

---

\(^1\)Bradley states the non-centrality parameter as \( \lambda = \delta^2/3 \) thus using a different definition than used here (and by R)
2.3 Psychometric functions for a selection of sensory discrimination protocols

2.3.2.4 Evaluation via the non-central $F$-distribution

To see why the triangle psychometric function can be evaluated via the non-central $F$-distribution we first need a few identities about $F$ and $\chi^2$ distributions (see e.g., [http://en.wikipedia.org/wiki/Noncentral_F-distribution](http://en.wikipedia.org/wiki/Noncentral_F-distribution)):

If $X^2 \sim \chi^2(\nu_1, \text{ncp} = \lambda)$ denotes a chi-square distributed random variable with $\nu_1$ degrees of freedom and non-centrality parameter (ncp) $\lambda$ and $Y^2 \sim \chi^2(\nu_2, \text{ncp} = 0)$, then

$$\tilde{F} = \frac{X^2/\nu_1}{Y^2/\nu_2} \sim F(\nu_1, \nu_2, \text{ncp} = \lambda).$$

Also, if $X \sim N(\mu, \sigma^2)$, then $X^2 \sim \sigma^2 \chi^2(\nu = 1, \text{ncp} = \mu^2/\sigma^2)$

Starting from equation (2.4), we have that $p_c = P(\tilde{F} > 3)$, where

$$\tilde{F} = \left( \frac{2|B - \frac{1}{2}(A_1 + A_2)|}{\sqrt{3|A_1 - A_2|}} \right)^2$$

Now write $\tilde{F}$ as

$$\tilde{F} = \frac{4}{3} \frac{Z^2}{Y^2}$$

where $Z = B - \frac{1}{2}(A_1 + A_2)$ and $Y = A_1 - A_2$. Also notice that since $A_1 + A_2$ is orthogonal to $A_1 - A_2$, $Z^2$ and $Y^2$ are independent. It follows that

$$Z \sim N(\delta, 3/2)$$
$$Y \sim N(0, 2)$$

hence

$$Z^2 \sim \frac{3}{2} \chi^2(\nu = 1, \text{ncp} = 2\delta^2/3)$$
$$Y^2 \sim 2\chi^2(\nu = 1, \text{ncp} = 0)$$

and

$$\frac{Z^2}{Y^2} \sim \frac{3}{4} F(\nu_1 = 1, \nu_2 = 1, \text{ncp} = 2\delta^2/3).$$

It follows that $\tilde{F} \sim F(\nu_1 = 1, \nu_2 = 1, \text{ncp} = 2\delta^2/3)$.

2.3.2.5 Examples in R

A straight forward brute force approach is to evaluate the integral in equation (2.3) with a general integration routine:
Sensory discrimination tests and Thurstonian models

> ## define parameter:
> delta <- 1
> ## integrand:
> int.fun <- function(x, delta) 2 * dnorm(x) *
> (pnorm(-x * sqrt(3) + delta * sqrt(2/3)) +
>  pnorm(-x * sqrt(3) - delta * sqrt(2/3)))
> ## perform integration:
> integrate(f=int.fun, lower=0, upper=Inf, delta = delta)
> 0.4180467 with absolute error < 3.5e-08

Alternatively we can evaluate the non-central F CDF:
> pf(q=3, df1=1, df2=1, ncp=(2 * delta^2)/3, lower.tail=FALSE)
> [1] 0.4180467

A speed comparison shows that evaluating the non-central F is more than a factor 10 faster and probably more accurate (the timings are standardized to refer to 1000 evaluations):
> system.time(replicate(1e4, integrate(f=int.fun, lower=0, upper=Inf, delta = delta))) / 10
>         user  system elapsed
> 0.184    0.000   0.186

> system.time(replicate(1e5, pf(q=3, df1=1, df2=1, ncp=(2 * delta^2)/3, lower.tail=FALSE))) / 1e2
>         user  system elapsed
> 0.0079   0.0000   0.0078

Computation via the multivariate expression is much slower than the univariate evaluations:
> system.time(replicate(1e3, {
>   p1 <- pmvnorm(lower=rep(-Inf, 2), upper=rep(0, 2),
>                 mean=mu, sigma=S, algorithm=TVPACK())
>   p2 <- pmvnorm(lower=rep(0, 2), upper=rep(Inf, 2),
>                 mean=mu, sigma=S, algorithm=TVPACK())
>   2 * c(p1 + p2)
> }))
>         user  system elapsed
> 1.080    0.000   1.111

2.3.2.6 Derivative of the psychometric function for the triangle protocol

Bi et al. (1997) states the derivative of the psychometric function for the triangle protocol as

\[ f'_{\text{triangle}}(\delta) = \sqrt{2/3} \phi(\delta/\sqrt{6}) \left[ \Phi(\delta/\sqrt{2}) - \Phi(-\delta/\sqrt{2}) \right] . \]
2.3 Psychometric functions for a selection of sensory discrimination protocols

Details about how this formula was reached are not given. Bi et al. (1997) refer to David and Trivedi (1962) but it is unclear whether more details are given in there.

2.3.3 Psychometric function for the unspecified method of tetrads

In the unspecified method of tetrads, an assessor is presented with four samples and instructed to group the samples in sets of two such that the samples within each group are most alike. The method of tetrads is less commonly used than the triangle protocol but has recently received more attention. The psychometric functions for the specified and unspecified tetrad protocols were considered by Ennis et al. (1998) and the power of the unspecified tetrad protocol was considered by Ennis and Jesionka (2011).

Let \( W \sim N(\delta, 1) \) denote a weak stimulus and \( S \sim N(0, 1) \) denote a strong stimulus, and let \( w_1, w_2 \) denote two independent random draws from \( W \) and similarly \( s_1, s_2 \) denote two independent random draws from \( S \). An assessor is then presented with the four samples \((w_1, w_2, s_1, s_2)\).

2.3.3.1 Expression via tri-variate normal distribution

A correct response is given if the groups are identified as \((w_1, w_2)\) and \((s_1, s_2)\). If an assessor seeks to minimize the (perceptual) distance between pairs in each group, then a correct response is obtained if

\[
\begin{align*}
C_1 & \quad w_1 < w_2, \ w_2 < s_1, \ w_2 < s_2 \\
C_2 & \quad w_2 < w_1, \ w_1 < s_1, \ w_1 < s_2 \\
C_3 & \quad w_1 > w_2, \ w_2 > s_1, \ w_2 > s_2 \\
C_4 & \quad w_2 > w_1, \ w_1 > s_1, \ w_1 > s_2 
\end{align*}
\]

Now observe that \( P(C_1) = P(C_2) \) and similarly \( P(C_3) = P(C_4) \). The probability of a correct response is therefore given as

\[
p_c = 2\{P(C_1) + P(C_3)\}
= 2\{P(W_1 < W_2, \ W_2 < S_1, \ W_2 < S_2) + P(W_1 > W_2, \ W_2 > S_1, \ W_2 > S_2)\}
= 2\{P(U_1 < 0, \ U_2 < 0, \ U_3 < 0) + P(U_1 > 0, \ U_2 > 0, \ U_3 > 0)\}
\]

where

\[
\begin{align*}
U_1 & = W_1 - W_2 \\
U_2 & = W_2 - S_1 \\
U_3 & = W_2 - S_2 
\end{align*}
\]
The variable transformation is given by the following Jacobian:

\[
J = \begin{bmatrix}
1 & -1 & 0 & 0 \\
0 & 1 & -1 & 0 \\
0 & 1 & 0 & -1 \\
\end{bmatrix},
\]

and the variance-covariance matrix of \( U \) is given as

\[
\]

\[
= \begin{bmatrix}
2 & -1 & -1 \\
-1 & 2 & 1 \\
-1 & 1 & 2 \\
\end{bmatrix},
\]

and \( E[U] = [0, -\delta, -\delta]^T \).

We may therefore express the probability of a correct answer as

\[
p_c = \int_{-\infty}^0 \int_{-\infty}^0 \int_{-\infty}^0 f_U(z) \, dz + \int_0^\infty \int_0^\infty \int_0^\infty f_U(z) \, dz,
\]

where \( f_U \) is the trivariate normal distribution of \( U \) with mean and variance-covariance matrix as defined above.

### 2.3.3.2 Examples in R

```R
> ## define parameters:
> delta <- 1
> ## expected value of U
> mu.trd <- c(0, -delta, -delta)
> ## Jacobian
> J.trd <- rbind(c(1, -1, 0, 0),
>                c(0, 1, -1, 0),
>                c(0, 1, 0, -1))
> ## variance-covariance matrix of U:
> (S.trd <- tcrossprod(J.trd))

[,1] [,2] [,3]  
[1,]  2  -1  -1 
[2,] -1   2   1 
[3,] -1   1   2
> ## probability of correct answer:
> 2 * c(pmvnorm(lower=-Inf, upper=rep(0, 3), mean=mu.trd, sigma=S.trd,
> algorithm=TVPACK()) +
> pmvnorm(lower=rep(0, 3), upper=Inf, mean=mu.trd, sigma=S.trd,
> algorithm=TVPACK()))

[1] 0.4938084
```
2.3 Psychometric functions for a selection of sensory discrimination protocols

2.3.3.3 Univariate expression of the psychometric function

The probability of the event $C_1$ may be written as

$$P(C_1) = P(W_1 < W_2, W_2 < S_1, W_2 < S_2)$$

$$= \int_{-\infty}^{\infty} P(W_1 < z, S_1 > z, S_2 > z, W_2 = z) \, dz$$

$$= \int_{-\infty}^{\infty} P(W_1 < z, S_1 > z, S_2 > z | W_2 = z) P(B = z) \, dz$$

$$= \int_{-\infty}^{\infty} P(W_1 < z) P(S_1 > z) P(S_2 > z) P(W_2 = z) \, dz$$

$$= \int_{-\infty}^{\infty} \Phi(z) \{1 - \Phi(z - \delta)\}^2 \phi(z) \, dz$$

Similarly we may write the probability of the event $C_3$ as

$$P(C_3) = P(W_1 > W_2, W_2 > S_1, W_2 > S_2)$$

$$= \int_{-\infty}^{\infty} P(W_1 > z, S_1 < z, S_2 < z, W_2 = z) \, dz$$

$$= \int_{-\infty}^{\infty} \{1 - \Phi(z)\} \Phi(z - \delta)^2 \phi(z) \, dz$$

It can be shown that $p_c$ may be expressed as (Ennis et al., 1998)

$$p_c = 1 - 2 \int_{-\infty}^{\infty} \phi(z) \left\{2\Phi(z)\Phi(z - \delta) - \Phi(z - \delta)^2 \right\} \, dz.$$
Mixed-effects models have proven a valuable class of models in so many areas of science and engineering where statistics are applied that they are now ubiquitous. Outside the normal linear framework evaluation of the likelihood function and optimization of it has, however, proven to be a considerable challenge and an active research area since the beginning of the 1990’s with the seminal papers by Schall (1991) and Breslow and Clayton (1993). Since then a wealth of estimation methods have been proposed and compared. Among the most celebrated methods are penalized quasi likelihood (PQL) (Schall, 1991; Breslow and Clayton, 1993; Goldstein, 1986, 1989, 1991), the Laplace approximation (LA) (Liu and Pierce, 1994; Pinheiro and Bates, 1995; Pinheiro and Chao, 2006; Skaug and Fournier, 2006; Doran et al., 2007), Gauss-Hermite quadrature (GHQ) (Anderson and Aitkin, 1985; Lesaffre and Spiessens, 2001; Borjas and Sueyoshi, 1994; Hedeker and Gibbons, 1994, 1996; Lee, 2000) and adaptive Gauss-Hermite quadrature (AGQ) (Liu and Pierce, 1994; Pinheiro and Bates, 1995; Pinheiro and Chao, 2006), simulation methods and MCMC methods, possibly combined with an EM algorithm as in MCEM (Monte Carlo EM) or as in SEM (Stochastic EM) (McCulloch, 1994; Chan and Kuk, 1997; McCulloch, 1997; Booth and Hobert, 1999; Millar, 2004). Naturally there are also Bayesian attempts closely linked with MCMC methods (Zeger and Karim, 1991; Karim and Zeger, 1992). Several monographs discuss mixed models and their computation, including (McCulloch and Searle, 2001; Fahrmeir and Tutz, 2001; Diggle et al., 2002; Skrondal and Rabe-Hesketh, 2004; Demidenko, 2004; Fitzmaurice
et al., 2009).

Two important classes of mixed-effects models outside the normal linear framework are the generalized linear mixed models (GLMMs) and (Gaussian) nonlinear mixed models (NLMMs). Computational methods for these two classes have developed partially independently of each other but with a significant overlap of methodology. Their synthesis: Generalized nonlinear mixed models seem much less frequent.

In this chapter estimation of generalized linear mixed effects models is discussed with emphasis on non-stochastic approximations to the likelihood function and models for binomial response. The focus will be on computational methods known as the Laplace approximation, Gauss-Hermite quadrature and adaptive Gauss-Hermite quadrature. This excludes stochastic methods and methods that cannot be formulated as an approximation to the likelihood function. Bayesian methods are also excluded due to our focus on the likelihood function. The three chosen methods are closely connected mathematically and computationally, so it makes sense to treat them together. They are also among the most widely implemented methods in statistical software packages and therefore of most interest to users of statistical methods.

Binomial-normal models are one of the most widely applied instances of the GLMMs and therefore of particular interest. While the logit link is often used in medical and biological applications, the probit link is used almost exclusively in applications in the social sciences and econometrics. PQL and Laplace approximations are known to be least accurate for paired binary data and GHQ has been reported unstable for binomial data with large denominators, so the most challenging cases for the computational methods that we consider are covered by the binomial models considered here. The AGQ approximation is generally believed to be numerically stable, the most accurate, but also the computationally most intensive and therefore with the longest estimation time. Another important GLMM not considered here is the Poisson-normal model much used in biological applications.

In addition to a general discussion of challenges in estimation of GLMMs, this chapter is also a precursor to the following chapter on estimation of cumulative link models and mixed model versions thereof. Both chapters provide some background for the implementation of CLMs and CLMMs in the ordinal package. The computational methods for estimating GLMMs are also applied in the estimation of CLMMs though a little more complicated due to the more complicated nature of the ordinal response relative to the univariate in GLMs.

This chapter begins with a (mathematical) outline of generalized linear models in section 3.1 to provide a background for the outline of Newton and Fisher scoring algorithms for parameter estimation in section 3.1.1 which again provides a base for explaining the conditional mode estimation used in two important approximations to the likelihood of GLMMs; the Laplace approximation and
3.1 Outline of generalized linear models

Generalized linear models (GLMs) can be fitted in various ways. In this section two popular and closely related methods are described, namely a Newton algorithm and Fisher scoring. The model and its likelihood with gradients and Hessian are described below whereas the actual algorithms are described in section 3.1.1. In section 3.1.2 the details of binomial models and their estimation are worked out for logit and probit links.

GLMs are models where the response follows a distribution in the exponential family including Poisson, binomial, gamma and Gaussian distributions. The expected value of the response is linked to a linear predictor, $\eta_i$ through a link function, $h(\cdot)$

$$E(Y_i|x_i) = \mu_i = h^{-1}(\eta_i), \quad \eta_i = x_i^T \beta$$

The distribution of the response is a member of the exponential family of distributions with log density of the form

$$\log p(y_i) = \frac{1}{\varphi} (y_i \theta_i - b(\theta_i)) + c(y_i, \varphi) \quad (3.1)$$

where the canonical parameter $\theta = \theta(\mu_i)$ is a function of the mean, $E(Y_i) = b'(\theta_i) = \mu_i$, $\text{Var}(Y_i) = \frac{\varphi}{w_i} b''(\theta_i) = \frac{\varphi}{w_i} V(\theta_i)$, where $V(\theta_i)$ is the variance function and $\varphi$ is an optional dispersion parameter. The term $c(y_i, \varphi)$ is a constant with respect to $\theta_i$ and ensures that the density integrates to one. The log-likelihood for $\beta$ can be written as

$$\ell(\beta, y_i) = w_i \log p(y_i) \quad (3.2)$$

where $w_i$ is a potential weight associated with the $i$th observation.

The gradient of $\ell(\beta, y_i)$ with respect to $\beta$, i.e. the score function, is

$$S(\beta, y_i) = \frac{\partial}{\partial \beta} \ell(\beta, y_i) = w_i \left[ y_i \frac{\partial \theta_i}{\partial \beta} - b'(\theta_i) \frac{\partial \theta_i}{\partial \beta} \right] = w_i \left[ \frac{\partial \theta_i}{\partial \beta} (y_i - \mu_i) \right] ,$$
where $\partial b(\theta_i)/\partial \beta = b'(\theta_i)\partial \theta_i/\partial \beta$ and
\[
\frac{\partial \theta_i}{\partial \beta} = \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} x_i \quad \text{since} \quad \mathbf{V}(\theta_i) = b''(\theta_i) = \frac{\partial b'(\theta_i)}{\partial \theta_i} = \frac{\partial \mu_i}{\partial \theta_i}.
\]
The term $\mathbf{V}(\theta_i)$ depends on the choice of $p(y_i)$ and $\partial \mu_i/\partial \eta_i$ depends on choice of $h(\cdot)$. If the canonical link is applied, then
\[
\theta_i = \theta(\mu_i) = h(\mu_i) = \eta_i, \quad \text{(3.3)}
\]
so $\partial \theta_i/\partial \beta = x_i$.

The Hessian, i.e. the second order derivative of $\ell(\beta, y_i)$ with respect to $\beta$ is
\[
H(\beta; y_i) = w_i \left[ \frac{\partial^2 \theta_i}{\partial \beta \partial \beta} (y_i - \mu_i) - \left( \frac{\partial \theta_i}{\partial \beta} \right)^2 b''(\theta_i) \right] \quad \text{(3.4)}
\]
where
\[
\frac{\partial^2 \theta_i}{\partial \beta \partial \beta} = \frac{\partial^2 \theta_i}{\partial \mu_i^2} \left( \frac{\partial \mu_i}{\partial \eta_i} \right)^2 x_i x_i^\top + \frac{\partial^2 \mu_i}{\partial \eta_i^2} \frac{\partial \theta_i}{\partial \mu_i} x_i x_i^\top
\]
The first term in (3.4) has expectation zero since $E(Y_i) = \mu_i$. When the canonical link function is applied, then, using (3.3):
\[
\frac{\partial^2 \theta_i}{\partial \beta \partial \beta} = \frac{\partial}{\partial \beta} \left( \frac{\partial \theta_i}{\partial \beta} \right) = \frac{\partial x_i}{\partial \beta} = 0
\]
so the first term in (3.4) vanishes, and $H(\beta; y_i) = \mathbf{E}(H(\beta; y_i))$.

The gradient and Hessian for all data are, assuming independence, given by
\[
S(\beta; y) = \sum_i S(\beta; y_i) \quad \text{and} \quad H(\beta; y) = \sum_i H(\beta; y_i)
\]

### 3.1.1 Estimation of generalized linear models

In this section Newton and Fisher scoring algorithms for the estimation of generalized linear models are outlined.

In matrix notation the gradient may be written
\[
S(\beta; y) = \mathbf{X}^\top \Psi^a \quad \text{(3.5)}
\]
where $X$ is the design matrix and in general $\Psi^a$ is an $n$-vector with elements
\[ \Psi^a_i = w_i(y_i - \mu_i)V(\theta_i)^{-1}\frac{\partial \mu_i}{\partial \eta_i} \]
but for canonical links
\[ \Psi^a_i = w_i(y_i - \mu_i). \]

The Hessian can be written as
\[
H(\beta, y) = X^\top \Psi^b X + X^\top \Psi^c X = X^\top (\Psi^b + \Psi^c) X \tag{3.6}
\]
where in general $\Psi^b$ and $\Psi^c$ are diagonal $n \times n$ matrices with elements
\[
\Psi^b_i = -w_i V(\theta_i)^{-1}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2,
\Psi^c_i = w_i(y_i - \mu_i) \left[\frac{\partial^2 \theta_i}{\partial \mu_i^2} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 + \frac{\partial^2 \mu_i}{\partial \eta_i^2} V(\theta_i)^{-1}\right]
\]
but for canonical links
\[
\Psi^b_i = -w_i V(\theta_i)
\Psi^c_i = 0.
\]

The Newton update reads
\[
\beta^{(i+1)} = \beta^{(i)} - H(\beta^{(i)}; y)^{-1} S(\beta^{(i)}; y)
\]
while the Fisher scoring update reads
\[
\beta^{(i+1)} = \beta^{(i)} - \mathbb{E}[H(\beta^{(i)}; y)]^{-1} S(\beta^{(i)}; y)
\]
and parenthesized superscripts denote iteration number. Observe that the Newton and Fisher scoring updates are identical when canonical links are applied.

### 3.1.2 Details for binomial models

The log-Bernoulli probability mass function reads
\[
\log p(y_i) = y_i \log \frac{\pi_i}{1 - \pi_i} + \log(1 - \pi_i) + c(y_i)
\]
where $y_i$ is the binary response. This is on the exponential family form (3.1) with elements $\theta_i = \log \frac{\pi_i}{1 - \pi_i}$, $\pi_i = \frac{\exp(\theta_i)}{1 + \exp(\theta_i)} = [1 + \exp(-\theta_i)]^{-1}$, $b(\theta_i) = -\log(1 - \pi_i) = \log(1 + \exp(\theta_i))$, $\frac{\partial b(\theta_i)}{\partial \theta_i} = b'(\theta_i) = \pi_i$. 


For Bernoulli, i.e. binary, observations $E(y_i|x_i) = \pi_i = \mu_i$, while for binomial observations with denominator $m_i$, $E(y_i|x_i) = m_i \pi_i = \mu_i$. The likelihood function for binomial observations can be obtained by using as response $y_i/m_i$, the ratio of success, in the Bernoulli probability mass function and $m_i$ as weights in (3.2).

For Bernoulli models we have that
\[
V(\theta_i)^{-1} = \frac{\partial \theta_i}{\partial \pi_i} = \frac{\partial}{\partial \pi_i} \log \left( \frac{\pi_i}{1 - \pi_i} \right) = [\pi_i(1 - \pi_i)]^{-1}
\]
and
\[
\frac{\partial^2 \theta_i}{\partial \pi_i^2} = \frac{\partial}{\partial \pi_i} \left( \frac{1}{\pi_i} + \frac{1}{1 - \pi_i} \right) = -\pi^{-2} + (1 - \pi)^{-2}
\]

Further, for the probit link
\[
\frac{\partial \pi_i}{\partial \eta_i} = \frac{\partial}{\partial \eta_i} \Phi(\eta_i) = \phi(\eta_i)
\]
and
\[
\frac{\partial^2 \pi_i}{\partial \eta_i^2} = \frac{\partial}{\partial \eta_i} \phi(\eta_i) = -\eta_i \phi(\eta_i),
\]
so
\[
\frac{\partial^2 \theta_i}{\partial \beta \partial \beta^\top} = x_i x_i^\top \left\{ \phi^2(\eta_i) \left[ -\pi_i^{-2} + (1 - \pi_i)^{-2} \right] - \frac{\eta_i \phi(\eta_i)}{\pi_i (1 - \pi_i)} \right\}.
\]

The gradient (3.5) and Hessian (3.6) are therefore identified with elements
\[
\Psi_i^a = w_i(y_i - \pi_i) \phi(\eta_i)[\pi_i(1 - \pi_i)]^{-1}
\]
\[
\Psi_i^b = -w_i[\pi_i(1 - \pi_i)]^{-1} \phi(\eta_i)^2
\]
\[
\Psi_i^c = w_i(y_i - \pi_i) \left\{ \phi(\eta_i)^2 \left[ -\pi_i^{-2} + (1 - \pi_i)^{-2} \right] - \frac{\eta_i \phi(\eta_i)}{\pi_i (1 - \pi_i)} \right\}
\]

For the logit link we have
\[
\frac{\partial \pi_i}{\partial \eta_i} = \frac{\partial}{\partial \eta_i} [1 + \exp(-\eta_i)]^{-1} = \frac{\exp(-\eta_i)}{[1 + \exp(-\eta)]^2}
\]
and
\[
\Psi_i^a = w_i(y_i - \pi_i)
\]
\[
\Psi_i^b = -w_i \pi_i(1 - \pi_i)
\]
\[
\Psi_i^c = 0
\]
3.2 Estimation of generalized linear mixed models

The root of the computation problem in GLMMs can be described as follows: Suppose we have observations from \( N \) clusters each with \( n_i \) observations such that \( y_{ij} \) is the \( j \)th observation on the \( i \)th cluster with \( i = 1, \ldots, N \) and \( j = 1, \ldots, n_i \). Suppose also that the random effects \( u_i \) for all \( i \) have a standard normal distribution with density, \( p(u_i) \) and that the distribution of the observations conditional on the random effects has density\(^\dagger\) \( p_\alpha(y_{ij} | u_i) \), where \( \alpha \) is the parameter vector including the variance, \( \sigma^2_u \) of the random effects.

The likelihood function is the joint density of the observations, \( p_\alpha(y) \) taken as a function of the parameters, \( \alpha \). This density is not directly available, but by standard rules of probability calculus it is given by

\[
p_\alpha(y) = \int p_\alpha(y,u) \, du = \int p(u)p_\alpha(y|u) \, du \tag{3.7}
\]

The log-likelihood can therefore be written as

\[
\ell(\alpha; y) = \log p_\alpha(y) = \log \int p(u)p_\alpha(y|u) \, du \tag{3.8}
\]

\[
= \sum_{i=1}^{N} \log \int p(u_i) \prod_{j=1}^{n_i} p_\alpha(y_{ij} | u_i) \, du_i \tag{3.9}
\]

where the last equality holds when observations from different clusters are assumed independent given the random effects.

The root of the computational challenge is the integral in (3.8), which, save for normal linear mixed models, does not have a closed form solution. Several approaches have been employed to overcome the integral: The integrand can be approximated by a function for which the integral has a closed form expression as in the Laplace approximation, the integral can be evaluated by numerical approximations, e.g. using Gauss-Hermite quadrature methods or by stochastic approximations. The accuracy of the last two methods can to some degree be improved by using more quadrature nodes or increasing the number of simulations. All these methods are based on the optimization of an approximate likelihood function. Other approximate methods such as the penalized quasi likelihood (PQL) method and relatives are defined as an algorithm and are not formulated as an approximation to an objective function.

Desirable features of a computational method are accuracy, numerical stability and estimation time. Sometimes we want results that are as accurate as possible, but we always want to know how accurate our results are for them to be

\(^\dagger\)Even though the distribution of the observations can be discrete, e.g. binomial or Poisson, we refer to the probability mass functions or probability density function of \( y_{ij} \) collectively as density.
scientifically meaningful. Generally there is a trade off between speed on one hand and accuracy and reliability on the other hand. Numerical stability is the ability of the method to yield the same accuracy for similar but slightly different data configurations. Predictable and understandable variations in accuracy are to some extent acceptable, but unpredictable oscillations in accuracy are not acceptable, because it would be impossible to know what the expected accuracy of the method would be on any particular data set. In the following, most emphasis will be given to estimation methods for models with a single scalar random effects term as these are easier coped with than the general case.

### 3.3 Some issues in evaluating properties of estimators

#### 3.3.1 Assessing the accuracy of approximations and estimators

Two types of accuracies are generally of interest: the accuracy of an approximation and the accuracy of an estimator. The former is mainly a computational issue and the latter is mainly a statistical issue. The statistical literature seems most occupied by the latter type of accuracy, but often the two are not clearly distinguished.

Let $\theta$ denote a true parameter, let $\theta_{ML}$ denote the ML estimator of $\theta$, let $\theta_X$ denote some other estimator, e.g. a REML-type estimator, and let $\theta_{X}^{Y}$ denote the $\theta_X$ estimator computed with computational approximation $Y$, where $Y$ could be one of PQL, LA, GHQ$_n$, AGQ$_n$, or some other computational method, and $n$ denotes the number of quadrature nodes.

In the literature, almost exclusively, one finds comparisons of $\theta_{Y}^{X}$ to $\theta$ for some values of $Y$ and $X$. This blurs the distinction between estimator and computational method and makes it difficult to tell whether any discrepancy between $\theta_{X}^{Y}$ and $\theta$ is due to bias in the estimator ($X$) or inaccuracy in the computational approximation ($Y$).

If $\theta$ is a variance parameter, e.g. $\sigma_u^2$, then $\theta_{ML}$ is not an unbiased estimator, especially not in situations where the information about $\theta$ is moderate and/or when $\theta$ is close to the boundary of its parameter space, i.e. when $\sigma_u = 0$. Bias in variance parameter estimators for example for the PQL estimator has been the focus of much research (e.g. Breslow and Lin (1995); Pawitan (2001b); Breslow (2003)). The accuracy of a computational method can be assessed by comparing $\theta_X$ to $\theta_{Y}^{X}$ where $Y$ is the computational method of interest. The accuracy of an estimator $X$ can be assessed by comparing $\theta_X$ to $\theta$. The latter requires that $\theta_X$ can be evaluated with sufficient accuracy, which can be an impediment.

A popular choice of estimator is the ML estimator, but for reasons of bias, par-
3.3 Some issues in evaluating properties of estimators

particularly in variance parameters, some statisticians prefer a REML-type estimator. The REML estimator in LMMs was proposed by (Patterson and Thompson, 1971) as the likelihood of residuals utilizing the orthogonality REML estimators of variance parameters in GLMMs have probably gained interest because the PQL algorithm involve an update of variance parameters by either a ML or REML estimator for an approximating LMM (Breslow and Clayton, 1993). Using the REML estimator in this intermediate step was believed to reduce bias in the PQL estimator of variance parameters.

There is however more to a statistical analysis than point estimation and the literature seems devoid of discussion and assessment of the accuracy of CIs. While Wald-based CIs can be reasonably precise for some regression parameters in GLMMs and NLMMs (but not always as the Hauck-Donner effect (Hauck Jr. and Donner, 1977) attests, see further in chapter 5), it seems that Wald CIs are often grossly imprecise for variance parameters (Pawitan, 2000).

Here we will focus entirely on genuine likelihood methods and particularly on the accuracy of computational approximations to the ML estimator, \( \hat{\theta}_{ML} \). \( \hat{\theta}_{AGQ} \) is numerically equivalent to \( \hat{\theta}_{ML} \) for high enough \( n \) and can be used as a standard against which other approximations can be compared. Properties of the ML estimator can be derived from the comparison of \( \theta \) used to generate the simulations and the estimated \( \hat{\theta}_{ML} \).

3.3.2 Asymptotic properties of estimators for binomial and multinomial observations

If the response \( Y \) is binomial and the covariates \( x \) are discrete, then there is a finite and relatively small number of distinct possible values of \( (y_{ij}, x_{ij}) \). Denote these possible sets by \( k = 1, \ldots, K \). Let \( \ell_{(k)}^Y \) denote the approximation to the log-likelihood function for computational method \( Y \) to the \( k \)th set. Further, let \( p_{(k)} \) denote the probability with which the \( k \)th set occurs, then the limiting log-likelihood approximation reads

\[
\ell_{(lim)}^Y = \sum_{k=1}^{K} p_{(k)} \ell_{(k)}^Y
\]

and the limiting \( \hat{\theta}_{MLE}^Y \) estimator is the maximizer of \( \ell_{(lim)}^Y \).

This avoids the use of time consuming Monte Carlo simulations to obtain \( \hat{\theta}_{MLE}^Y \) and provides it with any desired accuracy. Observe that \( \ell_{(lim)}^Y \) is the likelihood for all \( K \) data sets with \( p_{(k)} \) as weights, so basically standard estimation routines for GLMMs can be used directly.

Probably the simplest case is that of paired binary data. Assume that \( x_{ij} \in \{0,1\} \) is a treatment indicator variable, \( y_{ij} \in \{0,1\} \) is the binary response
and \( n_i = 2 \), then there are four possible response patterns \( (K = 4) \) whose distribution depends on the model.

Confidence intervals can also be assessed with this method, for example the limiting profile likelihood curves can be drawn, potentially with the limiting Wald approximation.

The same idea applies to Poisson count data where sets with \( p(k) < \varepsilon \) for some small \( \varepsilon \) are ignored. It also applies to cumulative link mixed models and other multinomial type models.

Joe (2008) used this approach to compare \( \theta_{\text{LA MLE}} \) to \( \theta \) with mentioning of \( \theta_{\text{AGQ MLE}} \). There is, however, no explicit comparison of approximation \( \theta_{\text{LA MLE}} \) to \( \theta_{\text{MLE}} \), nor of the accuracy of the ML estimator \( \theta_{\text{MLE}} \) to \( \theta \).

Monte Carlo simulations may still be needed to evaluate the robustness of the computational methods toward unbalance, outliers, starting values etc., and possibly also to evaluate average estimation times.

### 3.3.3 Comparison of approximations and estimators

There are many choices to be made during implementation of computational methods and probably different software houses have made different choices. In commercial software packages the details of the implementations are not accessible and the values of tuning parameters may not be publicly available. This hampers comparison of computational methods since only specific, but generally unknown implementations are being compared.

The properties of an estimation method depend not only on the properties of the integral approximation, but also on how the likelihood function is optimized and how convergence is judged. Some papers describe algorithms particular to their formulation (Wolfinger and Lin, 1997; Raudenbush et al., 2000; Hedeker and Gibbons, 1994). The properties of these algorithms are in general unknown. It may be that the algorithms, due to alternating steps or other approximations, lead to estimators that are not the maximizers of the approximated likelihood. We advocate general purpose nonlinear quasi-Newton optimizers (Nielsen, 2000; Nielsen and Mortensen, 2009; Nocedal and Wright, 2006) with accurate finite difference evaluations of the gradient when needed to optimize the approximated log-likelihood function. This ensures that the point of convergence is an optimum of the approximated log-likelihood function, although it may in general be a local optimum.

The choice of convergence criteria are important, not only because they can grossly affect the parameter estimates obtained, but also because they are important in identifying model fits that did not converge. Necessary conditions for convergence are a small gradient and positive definiteness of the Hessian matrix. These can be approximated via finite differences using Richardson’s extrapolation (Richardson, 1910; Richardson and Gaunt, 1927). See Eldén et al.
(2004) for an introduction to Richardson’s extrapolation and Gilbert (2009) for an implementation.

An accurate evaluation of the Hessian is also important in order to obtain accurate standard errors of the model parameters. While this is only rarely mentioned, some contributions propose to use the final BFGS-updated Hessian from a quasi-Newton optimization (e.g. Joe, 2008). This Hessian depends strongly on the choice of starting values and is in general an inaccurate approximation to the true Hessian.

### 3.4 Computational methods for generalized linear mixed models

In this section three important computational methods for GLMMs are described: the Laplace approximation, the Gauss-Hermite quadrature (GHQ) and adaptive Gauss-Hermite quadrature (AGQ). The relation to some approximate methods, such as penalized quasi likelihood and the h-likelihood approaches are also mentioned.

#### 3.4.1 The Laplace approximation

The Laplace approximation (Tierney and Kadane, 1986; Barndorff-Nielsen and Cox, 1979, 1989) was suggested for estimation in NLMMs by Pinheiro and Bates (1995, 2000). It was also considered for GLMMs by Liu and Pierce (1993, 1994) and further developed for nested random effect structures (multilevel models) by Pinheiro and Chao (2006) for canonical links. The accuracy of the LA for binomial and Poisson GLMMs and cumulative link mixed models, in particular the proportional odds models with random effects, was studied by (Joe, 2008).

The Laplace approximation corresponds to the approximation of the log-integrand by a quadratic function for which the integral has an analytical solution:

\[
\ell(\alpha; y) = \log \int \exp \{ \log p_\alpha(y, u) \} \, du \\
\approx \ell_{LA}(\alpha, y) = \log \int \exp \{ t(\alpha, u; y) \} \, du \\
\ell_{LA}(\alpha, y) = \log p_\alpha(y, \hat{u}) + \frac{q}{2} \log(2\pi) - \frac{1}{2} \log |D(\alpha, \hat{u})|
\]
where

\[
\log p_\alpha(y, u) = \log p_\alpha(y|u) + \log p(u)
\]

\[
t(\alpha, u; y) = \log p_\alpha(y, \hat{u}) - \frac{1}{2} (u - \hat{u})^T D(\alpha, \hat{u})(u - \hat{u})
\]

\[
D(\alpha, \hat{u}) = - \left. \frac{\partial^2 \log p_\alpha(y, u)}{\partial u \partial u^T} \right|_{u=\hat{u}}
\]

\[
\hat{u} = \arg \max_u \log p_\alpha(y, u)
\]

so \(t(\alpha, u; y)\) is the second order Taylor approximation of the joint log density \(\log p_\alpha(y, u)\) in \(u\) around the mode \(\hat{u}\) and \(D(\alpha, \hat{u})\) is the negative Hessian evaluated at the mode.

The accuracy of the LA therefore depends on the closeness of the joint log-density to a quadratic function, or equivalently, the closeness of the joint density to a Gaussian function. Since the marginal log-density of \(u\) is exactly quadratic in \(u\), the accuracy of the LA depends on the closeness of \(\log p_\alpha(y|u)\) to a quadratic function in \(u\).

If \(\log p_\alpha(y|u)\) is a binomial log-density, then the closeness to a quadratic function increases with the binomial denominator and with the closeness of \(\pi\) to 1/2. For probit and logit links \(\pi = 0.5\) when the linear predictor (cf. appendix 3.1.1) \(\eta = 0\), so large absolute values of \(\eta\) lead to less accuracy. For a single scalar random effects term we may for the \(j\)th observation on the \(i\)th cluster write \(\eta_{ij} = x_{ij}^T \beta + \sigma u_i\), so the accuracy of the LA decreases with the absolute size of \(x_{ij}^T \beta\) and with the size of \(\sigma\). The decrease of the accuracy of the LA with the size of \(\sigma\) is expected from published simulation studies, while the dependency of the accuracy on the mean structure, \(x_{ij}^T \beta\) does not seem to have been studied.

It seems appropriate to study the accuracy of the LA as a function of the size of the linear predictor or, equivalently, the size of the fitted probabilities.

For GLMMs the LA can be written (cf. eq. (3.11) in section 3.4.1.1)

\[
\ell_{LA}(\alpha; y) = \log p_\alpha(y|u) - \frac{1}{2} u^T u - \frac{1}{2} \log |I_q - V^T \Psi b V - R| \quad (3.10)
\]

Pinheiro and Chao (2006) considered GLMMs with canonical link functions where the \(R\) term in eq. (3.10) vanishes (cf. section 3.1.1). Doran et al. (2007) also considers the LA for GLMMs and (appear to) ignore the \(R\) term by referring to standard GLM weights in the Fisher scoring estimation of the random effect modes effectively using the expected Hessian rather than the observed. Pinheiro and Bates (1995) similarly used the expected Hessian to estimate the random effect modes for NLMMs.

The effect of using the expected rather than the observed Hessian in the LA has not yet been studied in the literature for neither NLMMs nor GLMMs to the best knowledge of the author. For NLMMs it can be argued that the difference
between the full and expected Hessian is probably small (Pinheiro and Bates, 1995; Bates and Watts, 1980), they are even identical when the random effects appear linearly in the model function.

Shun and McCullagh (1995); Shun (1997) and Raudenbush et al. (2000) consider higher order Laplace approximations. Although more accurate than the ordinary LA, the generalisability to complex design structures has not been investigated. For models where AGQ apply, it has not been thoroughly investigated which estimation method is the fastest. Further, Shun and McCullagh (1995); Shun (1997) showed by asymptotic arguments that the error of the ordinary LA considered above does not diminish as the sample size increase in some models where the number of random effects also increase. We are not aware of any numerical assessments of this; it may be that the LA is an adequate approximation in many practical situations despite the lack of asymptotic arguments for its validity.

3.4.1.1 Conditional mode estimation in GLMMs

Efficient application of the Laplace approximation is conditional on a speedy and robust estimation of the conditional mode of the random effects. In this section, drawing on methods for estimation of GLMs, the estimation of conditional modes is outlined.

In a GLMM the conditional distribution of the response given the random effects has an exponential family distribution with density, \( p(y|B = b) \) and conditional mean satisfying

\[ E[y|B = b] = h(\eta), \quad \eta = X\beta + Zb \]

The marginal distribution of the \( q \)-dimensional random effects is multivariate normal:

\[ B \sim N(0, \Sigma). \]

The linear predictor can be written as

\[ \eta = X\beta + Vu, \]

where \( V = Z\Lambda, \) \( \Lambda \) is the Cholesky factor of \( \Sigma \) such that \( \Lambda\Lambda^\top = \Sigma \) and \( U \) is standard multivariate normal, \( U \sim N(0, I_q). \)

The joint log density is

\[ \log p_\alpha(y, u) = \log p_\alpha(y|u) + \log p(u), \]

where

\[ \log p(u) = -r\log(2\pi)/2 - u^\top u/2 \]

By analogy with the development in section 3.1.1, the gradient with respect to \( u \) in the joint log density is

\[ S(u; \alpha, y) = V^\top \Psi^\alpha - u, \]
and the Hessian is

\[ H(u; \alpha, y) = V^\top \Psi^b V + R - I_q \]  \hspace{1cm} (3.11)

where

\[ R = V^\top \Psi^c V \]

and \( \Psi^a, \Psi^b, \Psi^c \) are described in section 3.1.1 and worked out for binomial models in section 3.1.2.

### 3.4.1.2 Random effects estimation and ridge regression

The computational problem of the conditional mode estimation is closely related to ridge regression (e.g. Hastie et al., 2001, p.60):

\[ \hat{\beta}_{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top y \]

where the tuning parameter, \( \lambda \) is related to the size of the variance parameter, \( \sigma_u \). The inflation of the diagonal of \( V^\top \Psi^b V \) by \( I_r \) ensures the positive definiteness of \( D \) and makes the computational problem well defined.

### 3.4.1.3 The link between LA, PQL and the \( h \)-likelihood

The PQL and \( h \)-likelihood methods are both connected to the LA and in some way approximations to the LA. The PQL method is motivated from the Laplace approximation by Breslow and Clayton (1993) and they also ignore the \( R \)-term. The PQL estimates have not been shown to be the maximizers of a single objective function. Instead, Schall (1991) and Breslow and Clayton (1993) show that the estimates can be obtained by iteratively applying estimation methods for LMMs. Usually the REML-method is employed for estimation of the variance components. The estimators for the fixed effects parameters, \( \beta \) and the random effects \( u \) are the joint maximizers of the PQL

\[ \log p_\alpha(y|u) + \log p(u) \]

which is the LA ignoring the last term. This could be appropriate if neither \( \Psi^b \), nor \( R \) depend much on \( \beta \). There do not seem to be any quantitative assessment of this dependency in the literature.

The variance parameters are then estimated from the REML likelihood for LMMs:

\[ -\frac{1}{2} \log |V| - \frac{1}{2} \log |X^\top V^{-1} X| - \frac{1}{2} (Y - X\hat{\beta})^\top V^{-1} (Y - X\hat{\beta}) = \frac{1}{2} \log |\Sigma_{\epsilon}^{-1} + Z^\top \Sigma_u Z| + \frac{1}{2} \log |\Sigma_u| \]

where \( V = \Sigma_{\epsilon}^{-1} + Z^\top \Sigma_u Z \), \( \Sigma_u \) is the covariance matrix of the marginal distribution of \( u \) and \( \Sigma_{\epsilon} \) is the covariance matrix of the approximated normal
distribution for the conditional distribution of the response given the random effects.

The PQL is identical to the $h$-likelihood as defined by Lee and Nelder (1996, p. 621) although the $h$-likelihood estimator of variance parameters differ from the PQL estimator. Due to the dependence (non-orthogonality) between variance parameters and regression parameters, the $h$-likelihood estimator of the regression parameters is also different from the PQL estimator. In later papers, (e.g. Lee and Nelder, 2000, 2003, 2004, 2005, 2006; Lee et al., 2006, 2007) these authors describe a range of $h$-likelihood estimators with various adjustments or corrections. Noh and Lee (2007) adjust the PQL aka $h$-likelihood as defined above by $\frac{1}{2}\log\left|\mathbf{V}_\tau^T \mathbf{Ψ}_b \mathbf{V}_\tau + \mathbf{I}_r\right|/(2\pi)$ essentially obtaining the modified LA for $\beta$ although the variance parameters are not obtained as the maximizers of the LA.

3.4.2 Gauss-Hermite quadrature

Standard, i.e. non-adaptive Gauss-Hermite quadrature (GHQ) is a method to approximate an integral by a finite weighted sum:

$$\int f(x) \exp(-x^2) \, dx \approx \sum_{h=1}^{N_{GHQ}} w_h f(x_h),$$

where the nodes, $x_h$ are roots of the $N_{GHQ}$’th order Hermite polynomial with associated weights, $w_h$. These can be found by algorithms described by Golub and Welsch (1969); Golub (1973) or from tables in Abramowitz and Stegun (1972). The weights satisfy $\sum_{h=1}^{N} w_h \equiv \sqrt{\pi}$ for all $N$. Pinheiro and Bates (1995) considered GHQ for NLMMs and found that it was unreliable. The accuracy of GHQ depends on the size of the variance parameter, and in discrete GLMMs, in contrast to NLMMs, the variance parameter is (loosely) bounded above. GHQ may therefore be unreliable in NLMMs, while reliable and accurate for at least some GLMMs. Since GHQ is computationally simpler and faster than AGQ, the method is of interest.

Gauss-Hermite quadrature is exact if the integrand is a normal density. Suppose

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{-\frac{(x - \mu)^2}{2\sigma^2}\right\} = \frac{1}{\sigma \sqrt{2\pi}} \exp(-x^*^2), \quad x^* = \frac{x - \mu}{\sqrt{2}\sigma}$$
then

\[
\int f(x) \, dx = \sqrt{2\sigma} \int \frac{1}{\sigma \sqrt{2\pi}} \exp(-x^2) \, dx^* \\
= \frac{1}{\sqrt{\pi}} \int \exp(-x^2) \, dx^* \\
\approx \sum_{h=1}^{N_{GHQ}} w_h \frac{1}{\sqrt{\pi}} \equiv 1
\]

for any number of nodes, so Gauss-Hermite quadrature is exact in this formulation—even with a single node and for a normal density with any valid mean and standard deviation. However, if we define \( f^*(x) = f(x) \exp(x^2) \), then we may write the quadrature rule as

\[
\int f(x) \, dx = \int f^*(x) \exp(-x^2) \, dx \\
\approx \sum_{h=1}^{N_{GHQ}} w_h f^*(x_h) \\
= \sum_{h=1}^{N_{GHQ}} w_h \exp(x_h^2) f(x_h)
\]

in which case the Gauss-Hermite quadrature is not exact for normal densities. With one quadrature node the approximation yields \( \sqrt{1/2} \) for the standard normal density. With ten quadrature nodes the approximation error is \( 1.24 \cdot 10^{-5} \), but the error quickly increases with departure from \( \mu = 0 \) and \( \sigma = 1 \). For example, if \( \mu = 1 \) and \( \sigma = .3 \), then the approximation error is 0.31.

For binomial-normal models (and GLMMs in general) the integrand has the form \( p(b_i) \prod_{j=1}^{n_i} p_{\alpha}(y_{ij}|b_i) \) cf. eq. (3.9) with linear predictor \( \eta_{ij} = x_{ij}^T \beta + b_i = x_{ij}^T \beta + \sigma u_i \) cf. section 3.4.1.1. We may therefore integrate out \( b_i \) with a Gauss-Hermite quadrature approximation as

\[
\int p(b_i) \prod_{j=1}^{n_i} p_{\alpha}(y_{ij}|b_i) \, db_i \approx \sum_{h=1}^{N_{GHQ}} w_h \exp(x_h^2) p(x_h) \prod_{j=1}^{n_i} p_{\alpha}(y_{ij}|x_h) \\
= \frac{1}{\sigma \sqrt{2\pi}} \sum_{h=1}^{N_{GHQ}} w_h \exp\{x_h^2 - x_h^2/(2\sigma^2)\} \prod_{j=1}^{n_i} p_{\alpha}(y_{ij}|x_h) .
\]
3.4 Computational methods for generalized linear mixed models

If instead we integrate out $u_i$ we get

$$\int p(u_i) \prod_{j=1}^{n_i} p_\alpha(y_{ij}|u_i) \, du_i \approx \sum_{h=1}^{N_{GHQ}} w_h \exp(x_h^2) \prod_{j=1}^{n_i} p_\alpha(y_{ij}|x_h)$$

$$= \frac{1}{\sqrt{2\pi}} \sum_{h=1}^{N_{GHQ}} w_h \exp(x_h^2/2) \prod_{j=1}^{n_i} p_\alpha(y_{ij}|x_h)$$

Alternatively, by changing the variable of integration to $x^*_i = b_i/(\sigma\sqrt{2})$, or equivalently $x^*_i = u_i/\sqrt{2}$, we may write the Gauss-Hermite quadrature approximation as

$$\int p(b_i) \prod_{j=1}^{n_i} p_\alpha(y_{ij}|b_i) \, db_i = \sigma\sqrt{2} \int \frac{1}{\sigma\sqrt{2\pi}} \exp(-x^*_i^2) p_\alpha(y_{ij}|b_i) \, dx^*_i$$

$$\approx \frac{1}{\sqrt{\pi}} \sum_{h=1}^{N_{GHQ}} w_h \prod_{j=1}^{n_i} p_\alpha(y_{ij}|b_h), \quad b_h = x_h\sigma\sqrt{2} \quad (3.12)$$

in which case it does not matter if we integrate out $b_i$ or $u_i$. The point is that despite the mathematical equivalence of the integrations, the actual formulations and implementations of GHQ can make a difference. There seems to be no quantitative examination of this in the literature.

Borjas and Sueyoshi (1994) observed that underflow can occur if, in formulation (3.12), $n_i$ is large and each $p_\alpha(y_{ij}|\cdot)$ is sufficiently small. The lower bound on floating point number representation in double precision is around $10^{-324}$ and the limit below which underflow may occur is around $10^{-308}$, i.e. roughly a factor $10^{16}$ larger than the lower bound. If $p_\alpha(y_{ij}|\cdot) = 1/2$ for all $j$, then underflow may occur with more than $308 \log 10 / \log 2 \approx 1023$ per cluster binary observations. If instead $p_\alpha(y_{ij}|\cdot) = .1$, only 308 per cluster observations will lead to underflow, while if $p_\alpha(y_{ij}|\cdot) = .9$, roughly 6731 per cluster observations are needed to cause underflow. More detailed examination of when underflow can occur and the consequences for likelihood approximations seem unavailable in the literature.

Lee (2000) proposed an algorithm to avoid underflow. He writes the GHQ approximation to the likelihood function in the form

$$\ell_{GHQ}(\alpha; y) = \sum_{i=1}^{N} \log \left\{ \frac{1}{\sqrt{\pi}} \sum_{h=1}^{N_{GHQ}} w_h \prod_{j=1}^{n_i} p_\alpha(y_{ij}|x_h\sigma\sqrt{2}) \right\}$$

and suggests to compute this as

$$\ell_{GHQ}(\alpha; y) = \sum_{i=1}^{N} \sum_{j=1}^{n_i} \log \left\{ \sum_{h=1}^{N_{GHQ}} p_\alpha(y_{ij}|x_h\sigma\sqrt{2}) \omega_{j-1,ih} \right\}$$
where the weights $\omega_{jih}$ are given by

$$\omega_{jih} = \frac{p_\alpha(y_{ij}|x_h\sigma\sqrt{2})\omega_{j-1,ih}}{\sum_{s=1}^{N_{GHQ}} p_\alpha(y_{ij}|x_h\sigma\sqrt{2})\omega_{j-1,is}} \tag{3.14}$$

and $\omega_{0ih} = w_h/\sqrt{\pi}$ for all $h$ and $i$. This estimation scheme effectively interchanges the inner product and sum. This is also a well known trick to avoid numerical underflow in the estimation of hidden Markov models (see e.g. Zucchini and MacDonald, 2009, p.46). Since the denominator of (3.14) is the $ij$th contribution to the likelihood function, the likelihood can be computed by the following algorithm

```plaintext
for $i = 1$ to $N$
do
  for $j = 1$ to $n_i$
do
    for $h = 1$ to $N_{GHQ}$
do
      compute $\omega_{jih}$
    end for
    store the denominator of (3.14) as $C_{ij}$
  end for
compute $\ell(\alpha; y) = \sum_{ij} C_{ij}$
```

This estimation scheme involves more computations than the direct evaluation of the log-likelihood function (3.13). The computational overload has, however, not been quantified.

Hedeker and Gibbons (1994, 1996) implement gradients and Hessian of the log-likelihood function using GHQ. Lesaffre and Spiessens (2001) remarks on the potential inadequacy in the approximations of the gradient and Hessian. The reported inaccuracy in the GHQ methods may therefore be due to other computational choices than the GHQ approximation to the likelihood function. Lesaffre and Spiessens (2001) and Rabe-Hesketh et al. (2005) find that GHQ is unreliable and gives biased estimates of variance parameters with large cluster sizes and larger variances. These authors also speculate that the reason for the failure of GHQ is that the integrand for a cluster contribution to the likelihood function is highly peaked and narrow, and so can fall, almost entirely, in between quadrature nodes.

Anderson and Aitkin (1985) is an early application of GHQ for the estimation of binomial-normal models. The paper contains a profile likelihood curve of a variance parameter clearly demonstrating the inappropriateness of a quadratic approximation to this.

### 3.4.3 Adaptive Gauss-Hermite quadrature

AGQ was proposed by Liu and Pierce (1994) as an improvement to GHQ. They noted that this method could prove valuable for GLMMs and remarked on the
connection to the LA. Pinheiro and Bates (1995) suggested AGQ for NLMMs and remarked on the connection to GHQ, the LA and motivated AGQ as the equivalent of importance sampling for GHQ. Pinheiro and Chao (2006) extended AGQ to multilevel GLMMs with canonical link functions. Liu and Pierce (1994) and Pinheiro and Bates (1995); Pinheiro and Chao (2006) shifted and scaled the quadrature nodes by the mode of the integrand and the Hessian at the mode. Rabe-Hesketh et al. (2005) and Naylor and Smith (1988) on the other hand shifted and scaled the quadrature nodes by the mean and the variance of the integrand. Rabe-Hesketh et al. (2005) also used AGQ to approximate the integrals defining the mean and variance of the integrand. They kept the location of the quadrature nodes fixed when evaluating the finite difference approximation to the gradient and Hessian for use in a Newton scheme for the estimation of the model parameters.

Extension of quadrature methods (GHQ and AGQ) to integrals of more than one dimension is difficult since the number of quadrature nodes increases rapidly. Rabe-Hesketh et al. (2005) proposed to use spherical quadrature rules (Stroud, 1971; Naylor and Smith, 1988) while Heiss and Winschel (2008) proposed to use a sparse grid integration rule (Smolyak, 1963; Gerstner and Griebel, 1998). Pinheiro and Chao (2006) used Cartesian quadrature for multilevel models exploiting the conditional independence structure of the problem.

Following Naylor and Smith (1982) and Liu and Pierce (1994) Gauss-Hermite quadrature can be re-expressed in terms of a normal density, \( \phi(t; \mu, \sigma) \) rather than \( \exp(-x^2) \):

\[
\int f(t) \phi(t; \mu, \sigma) \, dt = \int f(t) \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{ -\frac{(t - \mu)^2}{2\sigma^2} \right\} \, dt
\]

\[
= \frac{1}{\sqrt{\pi}} \int f(t) \exp(-z^2) \, dz, \quad t = \mu + \sqrt{2}\sigma z
\]

\[
\approx \sum_{h=1}^{N_{GHQ}} w_h \frac{f(t_h)}{\sqrt{\pi}}, \quad t_h = \mu + \sqrt{2}\sigma x_h
\]

For integration of \( g(t) \), let \( h(t) = g(t)/\phi(t; \mu, \sigma) \), so we may write

\[
\int g(t) \, dt = \int h(t) \phi(t; \mu, \sigma) \, dt
\]

\[
\approx \sum_{h=1}^{N_{GHQ}} w_h h(t_h), \quad t_h = \mu + \sqrt{2}\sigma x_h
\]

\[
= \sigma \sqrt{2} \sum_{h=1}^{N_{GHQ}} w_h \exp(x_h^2) g(t_h)
\]

subject to appropriate choice of the tuning parameters \( \mu \) and \( \sigma \). Liu and Pierce
(1994) suggested to take $\mu$ to be the mode of $g(t)$ and $\sigma = 1/D$, where

$$D = -\frac{\partial^2}{\partial t^2} \log g(t) \bigg|_{t=\mu}$$

Thus, if $g(t)$ is a Gaussian function, the quadrature rule is exact with a single node. Pinheiro and Bates (1995) suggested essentially the same approximation although using $E(D)$ rather than $D$ similar to their modification of the Laplace approximation, cf. section 3.4.1. There do not seem to be any quantitative assessment of the difference between using $E(D)$ or $D$ in the literature for GLMMs and NLMMs. This version of adaptive Gauss-Hermite quadrature is the Laplace approximation when a single node is used. For a reasonable number of nodes, e.g., ten, we may expect the quadrature rule to be insensitive to small differences in the choice of $\mu$ and $\sigma$. The main objective is that the integrand is reasonably covered by the quadrature nodes. The quadrature rule may therefore not be sensitive to the choice of expected versus observed Hessian, nor to whether the values of $\mu$ and $\sigma$ are updated for gradient evaluations. Perhaps even more approximate estimation of $\mu$ and $\sigma$ will be sufficient.

The AGQ approximation to the log-likelihood of a GLMM can be written as

$$\ell_{AGQ}(\alpha; y) = \sum_{i=1}^{N} \log \left\{ \sigma_i \sqrt{2} \sum_{h=1}^{N_{AGQ}} w_h \exp(x_h^2) p(t_{hi}) \prod_{j=1}^{n_i} p_{\alpha}(y_{ij}|t_{hi}) \right\}$$

$$= \sum_{i=1}^{N} \log \left\{ \frac{\sigma_i}{\sqrt{\pi}} \sum_{h=1}^{N_{AGQ}} w_h \exp(-t_{hi}^2/2) \prod_{j=1}^{n_i} p_{\alpha}(y_{ij}|t_{hi}) \right\}$$

(3.15)

where $t_{hi} = \mu_i + \sigma_i \sqrt{2} x_h$.

Rabe-Hesketh et al. (2005, p.303) states that “... adaptive quadrature is superior [to ordinary quadrature] since it requires fewer quadrature nodes.” in their discussion of GHQ and AGQ. It is not clear whether the (claimed) superiority is with respect to integration accuracy, computational speed, both or some other feature.

Approximating an integral with quadrature naturally takes less time the fewer the number of quadrature nodes. While AGQ often, but maybe not always, needs fewer nodes than GHQ to obtain the same accuracy, the shifting and scaling of the quadrature nodes used in AGQ need to be determined from the integrand and this takes time as well. Whether the more complicated process involved in AGQ takes longer or shorter time than do GHQ seems not to have been investigated from reading the literature. Such an assessment naturally depends on the particular implementation of the quadrature methods and for the AGQ how the measures of shift and scaling is obtained. It may be, however,
that there are cases where GHQ is unable to provide any reasonable accuracy irrespectively of how many nodes are used, while AGQ can work adequately.
This chapter is concerned with technical and computational aspects of cumulative link models with emphasis on the implementation in the ordinal package. Cumulative link models are introduced in appendix J and tutorials using the ordinal package are found in appendices K and L.

4.1 Cumulative link models

A cumulative link model (CLM) is a model for an ordinal response variable, $Y_i$ that can fall in $j = 1, \ldots, J$ categories where $J \geq 2$. A standard cumulative link model is

$$
\gamma_{ij} = F(\eta_{ij}), \quad \eta_{ij} = \theta_j - \mathbf{x}_i^\top \beta, \quad i = 1, \ldots, n, \quad j = 1, \ldots, J, \quad (4.1)
$$

where

$$
\gamma_{ij} = P(Y_i \leq j) = \pi_{i1} + \ldots + \pi_{ij}
$$

are cumulative probabilities\footnote{we have suppressed the conditioning on the covariate vector, $\mathbf{x}_i$, so we have that $\gamma_{ij} = \gamma_j(\mathbf{x}_i)$ and $P(Y_i \leq j) = P(Y \leq j|\mathbf{x}_i)$.}, $\eta_{ij}$ is the linear predictor and $\mathbf{x}_i^\top$ is a $p$-vector of regression variables for the parameters, $\beta$ without a leading column for an intercept. The inverse link function, $F$ is the CDF of a distribution in the location-scale family of distributions. Common choices of $F$ are the CDFs of
the logistic, normal, and Gumbel distributions. The thresholds or cut-points are strictly ordered:

\(-\infty \equiv \theta_0 \leq \theta_1 \leq \ldots \leq \theta_{J-1} \leq \theta_J \equiv \infty\).

In this chapter we describe the estimation of cumulative link models as it is approached in the R package ordinal. We begin by describing a matrix representation of cumulative link models. The matrix representation of cumulative link models is important in order to efficiently evaluate the log-likelihood and the first and second derivatives of the log-likelihood function with respect to the parameters. These are important in order to efficiently optimize the likelihood function when fitting cumulative link models.

### 4.2 Matrix representation of cumulative link models

Just as a linear model can be expressed as $Y = X\beta + \varepsilon$, cumulative link models can also be expressed in terms of matrices. The matrix representation of cumulative link models is a little more complicated than it is for linear models and (univariate) generalized linear models because of the special role of the threshold parameters.

The likelihood function for the cumulative link model in (4.1) can be written as

$$
\ell(\theta, \beta; y) = \sum_{i=1}^{n} w_i \log \pi_i,
$$

where $i$ index observations, $w_i$ are case weights and $\pi_i$ is the probability of the $i$th observation falling in the response category that it did, i.e. $\pi_i$ are the non-zero elements of $\pi_{ij}I(Y_i = j)$. For ungrouped observations $w_i = 1$ for all $i$ while for grouped observations $w_i$ is the observed frequency of the $i$th observation. Grouped observations can always be put in the ungrouped form, while genuinely ungrouped observations arise if, for example, each ordinal response is associated with a continuous covariate.

The probabilities, $\pi_i$ can be expressed as $\pi_i = \gamma_{i1} - \gamma_{i2}$ where $\gamma_{ik} = \gamma_{ij}$ for $j$ that satisfies $j = Y_i - k + 1$. Using the $k$-subscript notation, we can express the cumulative link model (4.1) as

$$
\gamma_{ik} = F(\eta_{ik}) , \quad \eta_{ik} = \theta_{ik} - x_i^T \beta, \quad i = 1, \ldots, n, \quad k = 1, 2 ,
$$

where $\theta_{ik} = \theta_j$ for $j$ that satisfies $j = Y_i - k + 1$. This re-expression of the CLM (4.1), motivates the following matrix formulation of cumulative link models:

$$
\gamma_k = F(\eta_k) , \quad \eta_k = A_k\xi - X\beta + o_k ,
$$

(4.2)
where the inverse link function, $F$ applies element wise to its arguments, $\xi$ denotes the central $q = J - 1$ elements of $\theta$ and $\beta$ is a $p$-vector of regression parameters for the design matrix $X$ of dimension $n \times p$. The design matrix $A$ is of dimension $n \times J$ such that $A_i^\top$ is a $J$-vector with a 1 at the $j$'th position and zeros elsewhere when $Y_i = j$ is observed. We define $A_1 = A(, −J)$ as $A$ without its $J$th column, similarly $A_2 = A(, −1)$, $o_1 = A(J, \theta_J)$, and $o_2 = A(1, \theta_0)$. The terms $o_k$ function as offsets in the predictors, i.e. fixed terms that do not depend on the parameters. They are computed under the definition $0 \cdot \infty \equiv \infty$ which means that some elements of $\eta_k$ are $\infty$, so it is assumed$^2$ that $F(−\infty) = 0$ and $F(\infty) = 1$.

Even more compact, we may write the model as

$$\gamma_k = F(\eta_k), \quad \eta_k = B_k \psi + o_k, \quad k = 1, 2,$$

(4.3)

where $\psi = [\xi^\top, \beta^\top]^\top$ is a $(q + p)$-vector, and $B_k = \partial \eta_k / \partial \psi = [A_k, −X]$ is a $n \times (q + p)$ matrix. The log-likelihood function for model (4.3) may be written

$$\ell(\psi; y) = w^\top \log \pi, \quad \pi = \gamma_1 - \gamma_2.$$

This expression of cumulative link models is the basis for extension of cumulative link models that we shall consider in the following as well as an efficient estimation scheme via a Newton algorithm.

### 4.3 A Newton algorithm for ML estimation of CLMs

The Newton algorithm is an iterative method that produce a sequence of estimates $\psi^{(0)}, \ldots, \psi^{(i)}, \ldots$. Here we used parenthesized superscripts to denote iterations. From the $i$th estimate the $(i + 1)$th estimate are given by

$$\psi^{(i+1)} = \psi^{(i)} - \alpha h^{(i)}, \quad h^{(i)} = H(\psi^{(i)}; y)^{-1} g(\psi^{(i)}; y)$$

where $h^{(i)}$ is the step of the $i$th iteration, $H(\psi^{(i)}; y)$ is the Hessian and $g(\psi^{(i)}; y)$ is the gradient of the log-likelihood function with respect to the parameters evaluated at the current estimates. The step factor, $\alpha$ is introduced to ensure that the Newton step causes an increase in the log-likelihood. The Newton step may be too large to increase the log-likelihood, so the step factor is halved successively until the step is acceptable. The gradient and Hessian are defined in the following section.

$^2$In the actual implementation we set $\theta_0 = −10^5$ and $\theta_J = 10^5$ to accomplish this behaviour.
4.4 Gradient and Hessian for standard CLMs

The gradient is the first derivative of the log-likelihood function with respect to the parameters. Using the chain rule we may write this as

\[ g(\psi; y) = \frac{d\ell(\psi; y)}{d\psi} = \frac{d\pi^\top}{d\psi} \frac{\partial \ell(\psi; y)}{\partial \pi} \]

where \( d \) is the full derivative operator and \( \partial \) is the partial derivative operator. The gradient, \( g(\psi; y) \) is a \((q+p)\)-vector while the factors in the product on the right hand side have dimensions \((q+p) \times n\) and \(n \times 1\).

Since \( \pi = \gamma_1 - \gamma_2 \) we may write the first factor of the gradient as

\[ \frac{d\pi^\top}{d\psi} = \frac{d\pi}{d\psi} = \gamma_1^\top - \gamma_2^\top \]

where \( \Phi_{1_k} \) for \( k = 1,2 \) are diagonal \( n \times n \) matrices with elements \( f(\eta_{ik}) \) for \( i = 1, \ldots, n \) and \( f = F' \) is the derivative of the inverse link function, \( F \). The second factor of the gradient is an \( n \)-vector with elements \( w_i / \pi_i \) that we denote \( \varpi \). In summary we write the gradient as

\[ g(\psi; y) = C^\top \varpi \]  

where we use \( C^\top = (d\pi^\top/d\psi) \) for easy reference.

The Hessian is the second order derivative of the log-likelihood function with respect to the parameters:

\[ H(\psi; y) = \frac{d^2\ell(\psi; y)}{d\psi^2} = \frac{\partial g(\psi; y)^\top}{\partial \psi} + \frac{d\pi^\top}{d\psi} \frac{\partial g(\psi; y)^\top}{\partial \pi}. \]

The Hessian, \( H(\psi; y) \) as well as both terms on the right hand side have dimension \((q+p) \times (q+p)\). The products of the second term have dimensions \((q+p) \times n\) and \(n \times (q+p)\). The first term on the right hand side can be written

\[ \frac{\partial g(\psi; y)^\top}{\partial \psi} = B_1^\top \Phi_2 B_1 - B_2^\top \Phi_2 B_2, \]

where \( \Phi_{2_k} \) are diagonal \( n \times n \) matrices with elements \( f'(\eta_{ik})w_i / \pi_i \) and \( f' = F'' \) is the second derivative of the inverse link function. The first factor of the second term of the Hessian, \((d\pi^\top/d\psi)\) is also part of the gradient and given in (4.4). The second factor of the second term can be identified as

\[ \frac{\partial g(\psi; y)^\top}{\partial \pi} = -\Phi_3 C, \]

where \( \Phi_3 \) is a diagonal \( n \times n \) matrix with elements \( w_i / \pi_i^2 \). In summary the Hessian may be written as

\[ H(\psi; y) = B_1^\top \Phi_2 B_1 - B_2^\top \Phi_2 B_2 - C^\top \Phi_3 C \]

(4.6)
4.5 Structured thresholds

The diagonal matrices are used here for notational convenience; the actual computations are performed element wise. This is much more efficient since computations with a lot of zeros are avoided.

Expressions for $F$, $f$ and $f'$ for the link functions are given in appendix J.

4.5 Structured thresholds

The only restriction on the thresholds, $\theta$ (equivalently $\xi$) is that they should be ordered increasingly. In some applications it is relevant to further restrict the thresholds to be symmetric or to be equidistant (equally spaced). See Christensen (2012), cf. appendix J.

Let a Jacobian, $J$ define the mapping between the thresholds, $\xi$ and the new set of threshold parameters, $\alpha$ such that $\xi = J\alpha$, where $n_\alpha \leq n_\xi$ ($n_\xi = q = J - 1$) with equality when $J$ merely represents a re-parameterization of the thresholds. See appendix 4.5.1 for examples of $J$.

The CLM may still be written as (4.3) if we define $\psi = [\alpha^\top, \beta^\top]^\top$, and $B_k = [A_k J, -X]$. The CLM in (4.3), the gradient, the Hessian and the Newton algorithm therefore applies to CLMs with structured thresholds with no other change than these modifications of the design matrices and parameter vectors.

4.5.1 Jacobians for structured thresholds

The particular form of $J$ depends on the assumption about the thresholds. The following examples of $J$ follows the definitions of symmetric and equidistant thresholds in (Christensen, 2012).

With $J = 6$ ordered response categories, the Jacobian assuming symmetric thresholds is

$$J = \frac{\partial \xi^\top}{\partial \alpha} = \begin{bmatrix}
1 & 0 & -1 \\
1 & -1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0 \\
1 & 0 & 1
\end{bmatrix},$$

so the first $\alpha$ parameter maps to the central threshold and remaining $\alpha$ parameters map to differences between the central and the remaining thresholds.

With an unequal number of response categories, e.g. $J = 5$, the Jacobian is

$$J = \begin{bmatrix}
1 & 0 & -1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 1
\end{bmatrix},$$
so the two first $\alpha$ parameters map the central two thresholds and the remaining $\alpha$ parameters denote distances from the central thresholds.

Equidistant thresholds can be fitted by using the following Jacobian, shown here for $J = 6$:

$$J^\top = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}.$$  

The first $\alpha$ parameter maps to the first threshold, and the second, i.e. last $\alpha$ represents the distance between consecutive thresholds. The standard flexible thresholds are obtained by taking $J$ to be the $n_\xi \times n_\xi$ unit-diagonal matrix.

### 4.6 Nominal effects

In the CLM (4.1) the thresholds are constant with respect to explanatory variables $X$. This assumption may be relaxed by letting the thresholds depend on one or more explanatory variables. For categorical explanatory variables (factors) we consider a model that contains a set of thresholds for each level of the explanatory variable. For continuous explanatory variables we consider models which contain $J - 1$ slopes rather than a single slope for all $j$.

Let $\hat{X}$ denote the design matrix for the factor $G$, where the first column is $1_n$ for the intercept, and the remaining $(n_g - 1)$ columns represent contrasts to the first level of $G$. We then define the design matrix for the new $n_\alpha n_g$-vector of threshold parameters, $\nu$ as

$$D_k = A_k : \hat{X}, \quad k = 1, 2,$$

where we use the notation $B : C$ to denote the matrix obtained by taking each column of $B$ and multiplying it element-wise by each column of $C$. Since the first column of $\hat{X}$ is $1_n$, the first $n_\alpha$ columns of $D_k$ are just $A_k$.

In this parameterization the first $n_\alpha$ elements of $\nu$ are thresholds for the first level of $G$. The thresholds for the second level of $G$ are the (vector) sum of the first $n_\alpha$ elements of $\nu$ and the next $n_\alpha$ elements of $\nu$. Observe that $D_k$ can be viewed as the design matrix that results from the interaction between a term that generates $A_k$ and the variable $G$.

If $\hat{X}$ is the design matrix for a continuous explanatory variable, then $\hat{X}$ has two columns with $1_n$ in the first column and the continuous explanatory variable in the second. Then $\nu$ is a $2n_\alpha$-vector where the first $n_\alpha$ elements are threshold parameters and the last $n_\alpha$ elements are response category specific slopes for the continuous variable.

A CLM with nominal effects may still be written as (4.3) if we define $\psi = [\nu^\top, \beta^\top]^\top$ and $B_k = [D_k, -\hat{X}]$. With these definitions the expressions for the log-likelihood, gradient and Hessian remain unchanged.
4.7 Cumulative link models for grouped continuous observations

In many applications where the response scale is ordinal, a continuous response scale could have been chosen instead. Rather than using, say, nine ordered and labeled categories in a study on the perceived intensity of some stimulus, the respondents could be asked to rate the degree of perceived intensity on a continuous scale from one to ten. The continuous response can, naturally, be coarsened into nine categories, or for that matter any other number of categories. From a statistical point of view, there is a marked difference between models for categorical variables and continuous variables, but that difference can be very weak, even close to non-existing, from a practical point of view. Consequently there is a challenge to smoothen the statistical distinction between models for categorical (ordinal) variables and continuous variables.

Suppose a variable \( S_i \sim N(\mu_i, \sigma^2) \) is mapped to a finite interval, \((a, b)\). If \( S_i \) is observed well within the interval, and, notably, no observations are recorded at the boundaries, then it may be reasonable to assume an ordinary linear model:

\[
S_i = \mu_i + E_i, \quad E_i \sim N(0, \sigma^2)
\]

Conceptually, this model is flawed because \( S_i \) is defined on the whole real line, while only a finite interval can be observed. Another model can be derived under the assumption that \( S_i \) is normally distributed. Suppose that \( S_i \) is not observed directly, but only a coarsened version, \( Y_i \) is observed falling in \( J \) categories such that \( Y_i = j \) is observed, if \( \tau_j \leq S_i \leq \tau_{j-1} \) and \( \tau_0 \leq \tau_1 \leq \ldots \leq \tau_{J-1} \leq \tau_J \) are equally spaced thresholds. This is a cumulative link model (4.1) with fixed, known thresholds. In this model, an intercept is identifiable, so we let \( X \) contain a leading column of ones and define \( \beta \) accordingly. Since the thresholds are fixed, the scale of the distribution is also identifiable, so we also let \( Z \) contain a leading column of ones. The model can be written as

\[
\eta_k = [-X\beta + o'_k] \exp(-Z\zeta) \quad (4.7)
\]

where \( o'_1 = A_1 \xi + o_1 = A\theta_{-0} \) and \( o'_2 = A_2 \xi + o_2 = A\theta_{-J} \).

One motivation for using a coarsened version of the continuous response rather than the raw observations is related to the amount of information in the perceived intensity of some stimulus. From a psychophysical point of view, we may be worried that recording the continuous response with several digits would be assigning too much information and precision to the observations. We would, perhaps, be inclined to record the observations with two, or perhaps only one significant digit. The latter conceptually corresponds to using category ranks
of a coarsened version of the continuous response as the response variable in an ordinary linear model. Regression on the category ranks implicitly assumes that the response scale is linear, i.e. that the distance between category ranks 1 and 2 represents the same difference in perceived intensity as the distance between category ranks 8 and 9. This may be an unreasonable restriction. Further, the definition of perceived intensity may differ between different respondents or groups of respondents, i.e. the response scale may mean different things to different people.

4.8 Likelihood for grouped continuous data

Continuous data are always recorded with finite precision and are therefore, in some sense, not continuous after all, but interval censored. Assume that a continuous stochastic variable, $X$ is assumed to follow a distribution with density $p_\theta(x)$ and is recorded with finite precision, $\varepsilon$, such that observing $x$ really means observing $x \in (x - \varepsilon/2, x + \varepsilon/2)$. The log-likelihood function can be written

$$\ell(\theta; x) = \log P_\theta\{X \in (x - \varepsilon/2, x + \varepsilon/2)\}$$

$$= \log \int_{x-\varepsilon/2}^{x+\varepsilon/2} p_\theta(x) \, dx$$

$$\approx \ell^*(\theta; x) = \log \varepsilon + \log p_\theta(x),$$

where the approximation improves as $\varepsilon \to 0$. Usually, in likelihood analyses, the approximate, or limiting likelihood function, $\ell^*$ is used and the factor, $\varepsilon$ ignored. Ignoring $\varepsilon$ is justified since only ratios of likelihoods are meaningful after all. The appropriateness of using $\ell^*$ rather than $\ell$ depends on the size of $\varepsilon$, since for small $\varepsilon$, there is no practical difference between $\ell$ and $\ell^*$.

The log-likelihood, $\ell$ is given by a CLM with fixed thresholds (4.7). Suppose $X_1, \ldots, X_N$ is observed in the interval $(a, b)$, then the number of (possible) data categories is $J = (b - a)/\varepsilon + 1$, and the end category boundaries are defined as $\tau_0 = a - \varepsilon/2$, and $\tau_J = b + \varepsilon/2$. The likelihood, $\ell$ can be adjusted by an amount $-\log \varepsilon$ to make it comparable to $p_\theta(x)$, which is usually reported as the likelihood for continuous models. Unfortunately this adjustment makes the likelihood incomparable between models fitted with unknown and fixed thresholds.

4.9 Implementation of CLMs in ordinal

We implement the negative log-likelihood rather than the its positive counterpart and use this as our objective function. It is traditional in the optimization
literature to *minimize* rather than *maximize*. We follow this tradition for easy interface with general purpose optimizers.

The negative log-likelihood of cumulative link models is implemented as follows:

```r
clm.nll <- function(rho) { ## negative log-likelihood
  with(rho, {
    eta1 <- drop(B1 %*% par) + o1
    eta2 <- drop(B2 %*% par) + o2
    fitted <- pfun(eta1) - pfun(eta2)
    if(all(fitted > 0))
      -sum(wts * log(fitted))
    else Inf
  })
}
```

*clm.nll* takes as argument an environment (*rho*) which is assumed to contain the design matrices *B1* and *B2*, the parameter vector *ψ* in *par*, the offset terms *o1* and *o2* in *o1* and *o2*, the weights, *w* in *wts* and the inverse link function, *F* in *pfun*. If the fitted probabilities are not all positive, *clm.nll* returns Inf. This may happen if a set of parameters are tried in which the threshold parameters are not of increasing magnitude.

The gradient (of the *negative* log-likelihood) is implemented in

```r
clm.grad <- function(rho) { ## gradient of the negative log-likelihood
  with(rho, {
    p1 <- dfun(eta1)
    p2 <- dfun(eta2)
    wtpr <- wts/fitted
    -crossprod(dpi.psi, wtpr)
  })
}
```

Here *dfun* is the derivative of the inverse link function, *f = F'* corresponding to the density function. The computation follows the description in (4.5) where *dpi.psi* corresponds to *C’*. Observe that the computation is performed at the current value of the fitted probabilities, *eta1* and *eta2*, thus it is assumed that the parameters have not changed since *clm.nll* was called.

The Hessian (of the *negative* log-likelihood) is implemented in

```r
clm.hess <- function(rho) { ## hessian of the negative log-likelihood
  with(rho, {
    dg.psi <- crossprod(B1 * gfun(eta1) * wtpr, B1) -
             crossprod(B2 * gfun(eta2) * wtpr, B2)
    -dg.psi + crossprod(dpi.psi, (dpi.psi * wtpr / fitted))
  })
}
```

where *gfun* is the function *f' = F''* — the gradient of the density function. Again this computation depends on computations in *clm.nll* and *clm.grad*.
the results of which are saved in the environment \( \mathbf{rho} \). The computations follow (4.6) to the extent that the diagonal matrices are avoided and vectorized computations are exploited instead. The term \( \mathbf{d}_{g, \psi} \) corresponds to the first two terms in (4.6).

### 4.10 Including scale effects in CLMs

Cumulative link models with scale or dispersion effects are of the form

\[
\gamma_{ij} = F(\eta_{ij}), \quad \eta_{ij} = \frac{\theta_j - \mathbf{x}_i^\top \beta}{\exp(\mathbf{z}_i^\top \mathbf{\zeta})},
\]

where \( \mathbf{z}_i^\top \) is an \( n_\zeta \)-vector of explanatory variables for the scale parameters \( \mathbf{\zeta} \). The scale effects can be thought of as modeling the scale of the latent distribution determined by \( F \)—hence the name. Similarly \( \mathbf{x}_i^\top \beta \) models the location of the latent distribution. Observe that the usual link functions are all members of a class of distributions known as location-scale distributions.

In matrix notation we may write

\[
\gamma_k = F(\eta_k), \quad \eta_k = \mathbf{Y}(\mathbf{B}_k \mathbf{\psi} + \mathbf{o}_k), \quad k = 1, 2,
\]

where \( \mathbf{Y} \) is a diagonal \( n \times n \) matrix with the elements of \( \exp(-\mathbf{Z} \mathbf{\zeta}) \) in the diagonal.

The gradient of a cumulative link model with scale effects can be written

\[
g(\psi, \mathbf{\zeta}; \mathbf{y}) = [\ell'_\psi(\psi, \mathbf{\zeta}; \mathbf{y})^\top, \ell'_\zeta(\psi, \mathbf{\zeta}; \mathbf{y})^\top]^\top
\]

The first element is similar to the gradient of a cumulative link model without scale effects:

\[
\ell'_\psi(\psi, \mathbf{\zeta}; \mathbf{y}) = \frac{d \pi^\top}{d \psi} \frac{\partial \ell(\psi, \mathbf{\zeta}; \mathbf{y})}{\partial \pi} = \mathbf{C}_2^\top \varpi,
\]

where \( \mathbf{C}_2^\top = \mathbf{B}_1^\top \Phi_{4_1} - \mathbf{B}_2^\top \Phi_{4_2} \) with \( \Phi_{4_k} \) being diagonal \( n \times n \) matrices for \( k = 1, 2 \) containing the elements of \( f(\eta_{ik}) \tau_i \) with \( \tau_i = \exp(-\mathbf{z}_i^\top \mathbf{\zeta}) \) and \( \varpi \) defined in section 4.4 as the \( n \)-vector with elements \( w_i / \pi_i \). The second element can be written

\[
\ell'_\zeta(\psi, \mathbf{\zeta}; \mathbf{y}) = \frac{d \pi^\top}{d \zeta} \frac{\partial \ell(\psi, \mathbf{\zeta}; \mathbf{y})}{\partial \pi} = \mathbf{C}_3^\top \varpi
\]

where \( \mathbf{C}_3^\top = -\mathbf{Z}^\top \Phi_5 \), \( \Phi_5 \) is diagonal with elements \( \eta_{i1} f(\eta_{i1}) - \eta_{i2} f(\eta_{i2}) \).

We may therefore write the gradient as

\[
g(\psi, \mathbf{\zeta}; \mathbf{y}) = [\mathbf{C}_2, \mathbf{C}_3]^\top \varpi.
\]
4.10 Including scale effects in CLMs

4.10.1 The Hessian of location-scale CLMs

The Hessian of location-scale models has the form

\[ H = \begin{bmatrix} D & E^\top \\ E & F \end{bmatrix} \]

where \( D \) is a \((q+p) \times (q+p)\) matrix, \( E \) is a \( n_\zeta \times (q+p) \) matrix and \( F \) is \( n_\zeta \times n_\zeta \) matrix.

Here \( D = \ell''_{\psi\psi}(\psi, \zeta; y) \) and given by

\[
D = \frac{d\ell''_{\psi\psi}(\psi, \zeta; y)}{d\psi} = \frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \psi} + \frac{d\pi^\top}{d\psi} \frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \pi}
\]

where

\[
\frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \psi} = B_1^\top \Phi_6, B_1 - B_2^\top \Phi_6, B_2 - C_2^\top \Phi_3 C_2
\]

and

\[
\frac{d\pi^\top}{d\psi} \frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \pi} = -C_2^\top \Phi_3 C_2,
\]

where \( \Phi_{6k} \) are diagonal \( n \times n \) matrices with elements \( \tau_i^2 w_i f'(\eta_{ik})/\pi_i \) and \( \tau_i = \exp(-z_i^\top \zeta) \). \( \Phi_3 \) is diagonal with elements \( w_i/\pi_i^2 \).

The element \( E^\top = \ell''_{\psi\psi}(\psi, \zeta; y) \) is given by

\[
E^\top = \frac{d\ell''_{\psi\psi}(\psi, \zeta; y)}{d\zeta} = \frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \zeta} + \frac{d\pi^\top}{d\zeta} \frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \pi}
\]

where

\[
\frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \zeta} = B_1^\top \Phi_7, Z - B_2^\top \Phi_7, Z - C_3^\top \Phi_3 C_2
\]

and

\[
\frac{d\pi^\top}{d\zeta} \frac{\partial \ell''_{\psi\psi}(\psi, \zeta; y)}{\partial \pi} = -C_3^\top \Phi_3 C_2,
\]

where \( \Phi_{7k} \) are diagonal with elements \(- [f(\eta_{ik}) + \eta_{ik} f'(\eta_{ik})] \tau_i w_i / \pi_i \).

The element \( F = \ell''_{\zeta\zeta}(\psi, \zeta; y) \) is given by

\[
F = \frac{d\ell''_{\zeta\zeta}(\psi, \zeta; y)}{d\zeta} = \frac{\partial \ell''_{\zeta\zeta}(\psi, \zeta; y)}{\partial \zeta} + \frac{d\pi^\top}{d\zeta} \frac{\partial \ell''_{\zeta\zeta}(\psi, \zeta; y)}{\partial \pi}
\]

where

\[
\frac{\partial \ell''_{\zeta\zeta}(\psi, \zeta; y)}{\partial \zeta} = Z^\top \Phi_8 Z,
\]

where \( \Phi_8 \) is diagonal with elements

\[
w_i / \pi_i \left\{ [\eta_{i1} f(\eta_{i1}) - \eta_{i2} f(\eta_{i2})]^2 / \pi_i - \eta_{i1} [f(\eta_{i1} + f'(\eta_{i1}) \eta_{i1})] + \eta_{i2} [f(\eta_{i2} + f'(\eta_{i2}) \eta_{i2})] \right\}.
\]
4.11 Flexible link functions

Aranda-Ordaz (1983) proposed a flexible link function:

$$F^{-1}_\lambda(\gamma_{ij}) = \log \left\{ \frac{(1 - \gamma_{ij})^{-\lambda} - 1}{\lambda} \right\},$$

which depends on the auxiliary parameter $\lambda \in ]0, \infty[$. When $\lambda = 1$, the logistic link function arise, and when $\lambda \to 0$,

$$\{(1 - \gamma_{ij})^{-\lambda} - 1\}/\lambda \to \log(1 - \gamma_{ij})^{-1},$$

so the log-log link arise. Thus, if $0 < \lambda < 1$, the density given by the inverse link function is left skewed, and if $\lambda > 1$, the density given by the inverse link function is right skewed.

The inverse link function and its derivative are given by

$$F(\eta) = 1 - (\lambda \exp(\eta) + 1)^{-\lambda^{-1}}$$
$$f(\eta) = \exp(\eta)(\lambda \exp(\eta) + 1)^{-\lambda^{-1} - 1}$$

In order to estimate $\lambda$ along with the other parameters in a quasi-Newton optimization, we need the gradient of the likelihood with respect to $\lambda$. Let

$$\ell'_\lambda(\psi, \zeta; \lambda; y) = \frac{\partial \pi^\top}{\partial \lambda} \frac{\partial \ell(\psi, \zeta, \lambda; y)}{\partial \pi} = \left[ \frac{\partial \gamma_1^\top}{\partial \lambda} - \frac{\partial \gamma_2^\top}{\partial \lambda} \right] w/\pi$$

where

$$\frac{\partial \gamma_k}{\partial \lambda} = \lambda^{-1} \exp(\eta_k)(\lambda \exp(\eta_k) + 1)^{-\lambda^{-1} - 1} - (\lambda \exp(\eta_k) + 1)^{-\lambda^{-1}} \log(\lambda \exp(\eta_k) + 1)\lambda^{-2}$$

4.12 Estimation of cumulative link mixed models

A cumulative link mixed model (CLMM) is a cumulative link model that contains random effects. A simple example of a cumulative link model is

$$\gamma_{tjh} = F(\theta_j - x_{th}^\top \beta - v_t)$$
$$t = 1, \ldots, q, j = 1, \ldots, J - 1, h = 1, \ldots, q_t,$$

where $v_t$ is the random effect for the $t$th group or cluster, $x_{th}$ are the explanatory variables for the $h$th sample in the $t$th cluster and $\beta$ are the parameters for the
fixed effects. Observe the minus before the random effects, so they have the same direction of effect as $\beta$. The random effects are assumed to be independently and identically normally distributed:

$$V_t \sim N(0, \sigma_v^2),$$

where $v_t$ is the realized value of the random variable $V_t$.

The ordinal observations are assumed to be multinomially distributed conditional on the realized values of the random effects, $V_t = v_t$. The likelihood function is mathematically equivalent to the density of the marginal distribution of the response, so we obtain the log-likelihood function by integrating the joint distribution of the response and the random effects over the random effects.

For the CLMMs with a simple scalar random effect as in (4.8), the log-likelihood function can be written

$$\ell(\theta, \beta, \sigma_v; y) = \log \int_{\mathbb{R}^n} p_{\theta, \beta}(y|u)p_{\sigma_v}(v) \, dv$$

$$= \sum_{t=1}^q \log \int_{\mathbb{R}} p_{\sigma_v}(v_t) \prod_{h=1}^{q_t} p_{\theta, \beta}(y_{th}|v_t) \, dv_t$$

where the last equality holds when observations from different clusters are assumed independent given the random effects.

In a more general notation we can write the cumulative link mixed models as:

$$\gamma_k = F(\eta_k) , \quad \eta_k = B_k \psi - Z v - o_k , \quad k = 1, 2 ,$$

(4.9)

where $Z$ is a $n \times q$ design matrix for the random effects, $v$ of length $q$. This notation accommodates any form of multiple scalar or vector random effects, being nested, crossed or partially crossed. The (marginal) distribution of the random effects is multivariate normal:

$$V \sim N(0, \Sigma_v) ,$$

where $\tau$ is the vector of unique variance parameters that parameterize the covariance matrix, $\Sigma_v$. We rewrite the CLMM in (4.9) in terms of unit random effects:

$$\gamma_k = F(\eta_k) , \quad \eta_k = B_k \psi - \dot{Z}_{\tau} u - o_k , \quad k = 1, 2 ,$$

(4.10)

where the unit random effects, $U$ has a multivariate standard normal distribution:

$$U \sim N(0, I) .$$

The random effects design matrix $\dot{Z}_{\tau}$ is defined as $\dot{Z}_{\tau} = Z \Lambda_\tau$, where $\Lambda_\tau$ is the Cholesky factor (“square root”) of $\Sigma_v$ such that $\Lambda_\tau \Lambda_\tau^\top = \Sigma_v$.

In the following exposition of computational methods for CLMMs we consider the model without scale effects for simplicity of notation. Including scale effects
in the models with a single random effect is straightforward and implemented in ordinal.

In the implementation of CLMMs in ordinal a general optimization routine is used to optimize the log-likelihood function or an appropriate approximation thereto. The main difficulty is in evaluating the log-likelihood sufficiently accurately and fast, hence these are the topics we will focus on in this section.

Following the outline of GLMMs in section 3.2 the log-likelihood can be written as

\[ \ell(\theta, \beta, \tau; y) = \log \int_{\mathbb{R}^n} p_{\theta, \beta, \tau}(y|u)p(u) \, du \]

Which in general does not reduce to a sum of uni-dimensional integrals.

In the following we outline three methods for estimation of CLMMs extending the outline of GLMM estimation in chapter 4. We will consider the Laplace approximation, Gauss-Hermite quadrature (GHQ) and adaptive Gauss-Hermite quadrature (AGQ) for a single random effect, and we will consider a multivariate Laplace approximation for general random effect structures.

### 4.12.1 Laplace approximation of CLMMs

Following the outline of the Laplace approximation in section 3.4.1, we write the Laplace approximation to the log-likelihood function for a CLMM as:

\[ \ell_{LA}(\psi, \tau; y) = \log p_{\psi, \tau}(y, \hat{u}) + \frac{q}{2} \log(2\pi) - \frac{1}{2} \log |D(\psi, \tau, \hat{u})| \]

\[ = w^\top \log \pi - \frac{1}{2} u^\top u - \frac{1}{2} \log |D(\psi, \tau, \hat{u})| \]

where \( D(\psi, \tau, \hat{u}) \) is the Hessian of \( \log p_{\psi, \tau}(y, u) = \log p_{\psi, \tau}(y|u) + \log p(u) \) with respect to \( u \) evaluated at \( \hat{u} = \arg \max_u \{ \log p_\alpha(y, u) \} \). Here

\[ \log p_{\psi, \tau}(y|u) = w^\top \log \pi, \quad \pi = \gamma_1 - \gamma_2 \]

with \( \gamma_k \) given in equation (4.9) and \( \log p(u) \) is the standard normal density of \( u \):

\[ \log p(u) = -\frac{q}{2} \log(2\pi) - \frac{1}{2} u^\top u. \]

We use a Newton procedure to estimate \( \hat{u} \) and therefore need the gradient and Hessian of \( \log p_{\psi, \tau}(y, u) \). The gradient is given by

\[ g_{\psi, \tau}(y, u) = \frac{d \log p_{\psi, \tau}(y, u)}{du} = \hat{Z}_\tau \phi - u \]

where \( \phi \) is an \( n \)-vector with elements

\[ \phi_i = -\{f(\eta_1i) - f(\eta_2i)\} w_i / \pi_i. \]
The Hessian is given by

\[ D(\psi, \tau, \hat{u}) = \hat{Z}_\tau^\top \Phi \hat{Z}_\tau - I_q \]  \hspace{1cm} (4.13)

where \( \Phi \) is an \( n \times n \) diagonal matrix with elements

\[ \Phi_i = -w_i \left\{ [f(\eta_{1i}) - f(\eta_{2i})]^2 / \pi_i - f'(\eta_{1i}) + f'(\eta_{2i}) \right\} \]  \hspace{1cm} (4.14)

The Newton update of the conditional mode of the random effects is then given by

\[ u^{(i+1)} = u^{(i)} - h, \]

where the Newton step \( h \) is the solution to \( D = hg \), which we solve in two stages using the Cholesky factorization of \( D = LL^\top \):

1. Solve \( Lc = g \) for \( c \)
2. Solve \( L^\top h = c \) for \( h \).

Upon estimation of the model parameters, the random effect estimates can be restored on the original scale with \( \hat{v} = \Lambda_{\tau} \hat{u} \).

The Newton procedure employed to estimate the conditional modes of the random effects involves the Hessian \( D \) which can be fairly large, so evaluating the gradient and Hessian directly as they are expressed in equations (4.11) and (4.13) is very costly. However, these matrices contain a lot of zeros as they arise from indicator matrices — they are said to be sparse matrices. This structure can be utilized by using matrix methods specially designed for sparse matrices and ordinal does this by interfacing to the Matrix R package (Bates and Maechler, 2012) that provides access to a C library for sparse matrix operations (Davis, 2006).

An additional improvement comes from realizing that the Hessian is not needed as such, rather the Cholesky factor of the Hessian, \( L \) is needed to compute the Newton step. The Matrix package provides a function that will update the Cholesky factor, \( L \) given \( \hat{Z} \) and \( \Phi \). This function takes advantage of the fact that since the pattern of zeros in \( \hat{Z} \) is constant, the pattern of zeros in the Cholesky factor is also constant, so it can avoid the entire analytical phase of the decomposition and only perform the numerical processes to update the non-zero values in the Cholesky factor.

### 4.12.1.1 Laplace approximation for CLMMs with a single random effect term

For a single random effect, the expressions for the gradient and the Hessian can be reduced to simple summations. Since these expressions are evaluated around 5 times for each evaluation of the log-likelihood, and the log-likelihood...
is evaluated hundreds of times during the course of optimizing the likelihood for parameter estimation, substantial speed improvements can be made by optimizing these evaluations. Such summations are not particularly fast in interpreted languages like R, and compiled languages like C can offer substantial speed improvements. Consequently these expressions are implemented C and the ordinal package links to this code.

For a CLMM with a single random effect terms, the \( t \)th element of the gradient can be written as

\[
g(\psi, \sigma_v, u_t) = \sum_{h=1}^{q_t} \sigma_v \phi_{th} - u_t
\]

and the \( t \)th diagonal element of the Hessian may be computed as

\[
D(\psi, \sigma_v, u_{tt}) = \sum_{h=1}^{q_t} \sigma_v^2 \Phi_{th} - 1
\]

where the elements of \( \phi \) and \( \Phi \) are given in equations (4.12) and (4.14) respectively.

The Laplace approximation in this case simplifies to

\[
\ell_{LA}(\psi, \sigma_u; y) = w^\top \log \pi - \frac{1}{2} u^\top u - \frac{1}{2} \sum_{t=1}^{q} \log D(\psi, \zeta, \tau, \hat{u}_{tt})
\]

### 4.12.2 Standard Gauss-Hermite quadrature

As discussed in section 3.4.2, standard, i.e. non-adaptive Gauss-Hermite quadrature (GHQ) is a method for approximating the integral of a function by a finite weighted sum:

\[
\int f(x) \exp(-x^2) \, dx \approx \sum_{s=1}^{N_{GHQ}} \omega_s \, f(x_s),
\]

where the nodes, \( x_s \) are roots of the \( N \)’th order Hermite polynomial with associated weights, \( \omega_s \). In the following we shall use other versions of the weights given by \( \omega_s^* = \omega_s \exp(x_s^2/2) \) and \( \omega_s^{**} = \log(\omega_s) = \log(\omega_s) + x_s^2/2 \).

The GHQ approximation to the marginal likelihood of a CLMM as implemented in ordinal can be written as

\[
\ell_{GHQ}(\psi, \sigma_v; y) = \frac{q}{2} \log(2\pi) + \sum_{t=1}^{q} \log \left[ \sum_{s=1}^{N_{GHQ}} \omega_s^* \prod_{h=1}^{q_t} \pi_{w_{th}}^{w_{th}} \right]
\]

\[
= \frac{q}{2} \log(2\pi) + \sum_{t=1}^{q} \log \left[ \sum_{s=1}^{N_{GHQ}} \exp \left\{ \omega_s^{**} + \sum_{h=1}^{q_t} w_{th} \log \pi_{th} \right\} \right].
\]
Here the quadrature nodes enter the CLMM in the following way:

$$
\gamma_{tjhs} = F(\theta_j - x_{th}^T \beta - \sigma_v x_s)
$$

with $t$, $j$ and $h$ as defined in connection with equation (4.8). As before, $\pi_{ths} = \gamma_{ths,j} - \gamma_{ths,j-1}$ and $w_{th}$ are prior weights. The latter formulation in eq. (4.15) is preferred to ensure numerical stability.

### 4.12.3 Adaptive Gauss-Hermite quadrature

Adaptive Gauss-Hermite quadrature (AGQ) shifts and scales the sum, so the function is sampled in the appropriate region (cf. section 3.4.3 for details):

$$
\int f(x) \exp(-x^2) \, dx \approx \sum_{s=1}^{N_{AGQ}} \omega_s^* f(x_s^*),
$$

where $\omega_s^* = K \omega_s$, $x_s^* = \hat{x} + K x_s$, $K = \sqrt{2/D}$, $D = \partial^2 \log f(x)/\partial x^2 |_{x=\hat{x}}$ and $\hat{x} = \arg \max_x f(x)$. AGQ for the likelihood of a CLMM with a single scalar random term (4.8) reads

$$
\ell_{AGQ}(\psi, \sigma_v; y) = \frac{q}{2} \log(2\pi) + \sum_{t=1}^{q} \left[ \log K_t + \log \sum_{s=1}^{N_{AGQ}} \exp(C_{ths}) \right]
$$

$$
C_{ths} = \log(\omega_s) + x_s^2 - (\hat{u}_t + K_t x_s)^2/2 + \sum_{h=1}^{q_t} w_{th} \log \pi_{ths}
$$

where $v_t = \sigma_v u_t$, $K_t = \sqrt{2/D(\psi, \sigma_v, \hat{u}_t)}$, and the quadrature nodes enter the CLMM in the following way:

$$
\gamma_{tjhs} = F\{\theta_j - x_{th}^T \beta - \sigma_v (\hat{u}_t + K_t x_s)\}.\]
Chapter 5

Likelihood ratio, score and Wald statistics

This chapter provides some technical background on the relations among the likelihood ratio (LR) (Wilks, 1938), score (Rao, 1948) and Wald (Wald, 1943) statistics.

An intuitive geometrical explanation of the Wald and score statistics as approximations to the LR statistic that we haven’t found in similar detail in the literature is provided here. Different versions of the score statistic relating to the choice of observed versus expected Fisher information matrix are discussed. The choice of information matrix in the score test seems not to have received the same level of attention as the choice has for the Wald test (cf. Efron and Hinkley, 1978; Pawitan, 2001a). A similar exposition of the relation among the three test statistics was given by (Buse, 1982) while the multiparameter case with nuisance parameters was not considered.

As an introductory example we consider score and Wald tests for a binomial parameter in section 5.1. In section 5.2 the Wald and score approximations to the LR statistic are worked out in one- and multi-parameter settings with nuisance parameters. As an illustration of the insight this geometrical explanation brings, the misbehaviour of the Wald statistic that can occur in logistic regression known as the Hauck-Donner effect (Hauck Jr. and Donner, 1977) is explained in section 5.3 by consideration of the shape of the log-likelihood function.
5.1 Score and Wald tests for a binomial parameter

Assume a random variable $X$ that follows a binomial distribution; $X \sim \text{binom}(\theta, n)$ with unknown fixed parameter $\theta$ and known sample size $n$. Let $x$ denote an observed value of $X$ and let $\theta$, the probability of “success” be the parameter of interest. The probability mass function for $X$ is given by

$$f_X(x; \theta, n) = \binom{n}{x} \theta^x (1 - \theta)^{n-x}$$

and the log likelihood is given by

$$\ell(\theta; x, n) = x \log \theta + (n - x) \log(1 - \theta) + C$$

where $C = \log \binom{n}{x}$ is a constant with respect to $\theta$ that we will ignore in the following. Further, $E[X] = n\theta$ and $V[X] = n\theta(1 - \theta)$.

Define the score function or gradient as

$$U(\theta; x, n) = \frac{\partial}{\partial \theta} \ell(\theta; x, n)$$

and the Hessian as

$$H(\theta; x, n) = \frac{\partial^2}{\partial \theta^2} \ell(\theta; x, n) .$$

Further define the observed Fisher information as

$$I(\theta; x, n) = -H(\theta, x, n)$$

and the expected Fisher Information as

$$\mathcal{I}(\theta; x, n) = E[I(\theta; x, n)] .$$

Note that these are functions of the unknown parameter and the observed data. For given data they may be evaluated at a particular value of the unknown parameter to produce a number.

Denote by $\theta_0$ the value of the parameter under the null hypothesis, and let $\hat{\theta} = x/n$ denote the maximum likelihood (ML) estimate of $\theta$.

For the binomial situation we have that:

$$U(\theta; x, n) = \frac{x - n\theta}{\theta(1 - \theta)}$$

$$H(\theta; x, n) = - \left[ \frac{x}{\theta^2} + \frac{n - x}{\theta(1 - \theta)} \right]$$

$$\mathcal{I}(\theta; n) = E \left[ \frac{x}{\theta^2} + \frac{n - x}{\theta(1 - \theta)} \right] = \frac{n}{\theta(1 - \theta)}$$
using that $E[x] = n\theta$. Observe that the expected information does not depend on the number of successes. Also observe that at the ML estimate we have the identity: $I(\hat{\theta}; x, n) = I(\hat{\theta}; n)$, while in general (for other $\theta$) $I(\theta; x, n) \neq I(\theta; n)$.

The variance of the ML estimator also denoted by $\hat{\theta} = X/n$ (this is a random variable) is given by the inverse expected Fisher information:

$$V[\hat{\theta}] = I(\hat{\theta}; n)^{-1} = \frac{\theta(1 - \theta)}{n}.$$

The familiar Wald statistic is obtained as the squared distance to the ML estimate divided by the variance at the ML estimate:

$$W(\theta_0; n) = \frac{(\hat{\theta} - \theta_0)^2}{\hat{\theta}(1 - \hat{\theta})/n}.$$

The score statistic is defined as

$$S(\theta_0; x, n) = \frac{U(\theta_0; x, n)^2}{I(\theta_0; x, n)}$$

and may for the binomial parameter be written as

$$S(\theta_0; x, n) = \frac{(x - n\theta_0)^2}{n\theta_0(1 - \theta_0)} = \frac{(x - n\theta_0)^2}{n\theta_0} + \frac{[(n - x) - n(1 - \theta_0)]^2}{n(1 - \theta_0)}$$

$$= \frac{(\hat{\theta} - \theta_0)^2}{\theta_0(1 - \theta_0)/n}$$

where the first formulation is probably the easiest to apply in practice, the second formulation makes the equivalence to Pearson’s statistic, $X^2 = \sum_j (o_j - e_j)^2/e_j$ explicit and the last formulation clarifies that the difference between the Wald and score statistic is whether the squared distance to the ML estimate is divided by the variance at the ML estimate (Wald) or at the null hypothesis (score).

Interestingly, when $\theta_0 = 1/2$, which is not uncommon, the score statistic simplifies to

$$S(\theta_0; x, n) = (2x - n)^2/n.$$

### 5.2 Contrasting the likelihood ratio, score and Wald statistics via the shape of the likelihood function

In this section we explain how three asymptotically equivalent statistics are related to each other. We do so by illustrating how they each relate to the
shape of the likelihood function. The score and Wald statistics can in this light be understood as approximations to the likelihood ratio statistic; the Wald statistic draws on information at the ML estimate and the score statistic uses information at the null hypothesis. We will use the binomial situation from the previous section as an illustration.

We start out with a single parameter setting and in section 5.2.1 a multiparameter setting is considered.

The likelihood ratio statistic is defined as

\[
LR = 2[\ell(\hat{\theta}; x, n) - \ell(\theta_0; x, n)]
\]

and just like the score and Wald statistics it asymptotically follows a \(\chi^2_1\) distribution under the null hypothesis.

The likelihood ratio statistic use information at both the ML estimate and at the null hypothesis, essentially the model has to be fitted at both parameter settings, while the Wald statistic use information only at the ML estimate, \(\hat{\theta}\), and the score statistic use information only at the null hypothesis, \(\theta_0\).

The Wald statistic can be understood via a quadratic approximation to the likelihood function. Define a second order Taylor approximation to the likelihood function as

\[
\tilde{\ell}(\theta, \theta_a) = \ell(\theta_a) + \ell'(\theta_a)(\theta - \theta_a) + \frac{1}{2}\ell''(\theta_a)(\theta - \theta_a)^2
\]

\[
\approx \ell(\theta_a) + \ell'(\theta_a)\theta - \theta_a + \frac{1}{2}\ell''(\theta_a)\theta - \theta_a)^2
\]

where the approximation is taken in the point \(\theta_a\) and considered as a function of \(\theta\). For notational convenience we have dropped mentioning of \(x\) and \(n\) as arguments.

The Wald statistic can be written as a likelihood ratio statistic where the log likelihood at the null hypothesis is based on the second order approximation taken in the ML estimate:

\[
LR_W = 2[\ell(\hat{\theta}) - \ell(\theta_0, \hat{\theta})] = 2[\ell(\hat{\theta}) - \ell(\hat{\theta}) - U(\hat{\theta})(\theta_0 - \hat{\theta}) + \frac{1}{2}I(\hat{\theta})(\theta_0 - \hat{\theta})^2] = I(\hat{\theta})(\theta_0 - \hat{\theta})^2
\]

since \(U(\hat{\theta}) \equiv 0\) by definition. Had we used the expected information rather than the observed information in the likelihood approximation, we would have obtained

\[
LR_W = I(\hat{\theta})(\theta_0 - \hat{\theta})^2 = (\theta_0 - \hat{\theta})^2 / \sqrt{\hat{\theta}}
\]
5.2 Contrasting the likelihood ratio, score and Wald statistics via the shape of the likelihood function

which more closely resembles the conventional definition of the Wald statistic. For the binomial parameter, however, both versions lead to the same statistic. The score statistic can also be understood via a quadratic approximation to the log-likelihood function. First we need to define \( \hat{\theta} \) as the value of \( \theta \) that solves

\[
\hat{\ell}'(\theta, \theta_0) = \frac{\partial}{\partial \theta} \hat{\ell}(\theta, \theta_0) = U(\theta_0) - \mathcal{I}(\theta_0)(\theta - \theta_0)
\]

such that

\[
\hat{\theta} = \theta_0 + U(\theta_0)/\mathcal{I}(\theta_0).
\]

This is seen to be the one-step Newton update of \( \theta_0 \) when \( \mathcal{I}(\theta_0) \) is used and the one-step Fisher scoring update of \( \theta_0 \) when \( \mathcal{I}(\theta_0) \) is used. We can understand \( \hat{\theta} \) as an estimate of, or an approximation to \( \hat{\theta} \).

The score statistic can be viewed as a likelihood ratio statistic where the log-likelihood at the ML estimate is based on the second order approximation taken under the null:

\[
LR_S = 2[\hat{\ell}(\hat{\theta}; \theta_0) - \ell(\theta_0)]
\]

\[
= 2[\ell(\theta_0) + U(\theta_0)(\hat{\theta} - \theta_0) - \frac{1}{2}\mathcal{I}(\theta_0)(\hat{\theta} - \theta_0)^2 - \ell(\theta_0)]
\]

\[
= 2U(\theta_0)(\hat{\theta} - \theta_0) - \mathcal{I}(\theta_0)(\hat{\theta} - \theta_0)^2
\]

\[
= 2U(\theta_0)/\mathcal{I}(\theta_0)U(\theta_0)/\mathcal{I}(\theta_0) - \mathcal{I}(\theta_0)U(\theta_0)^2/\mathcal{I}(\theta_0)^2
\]

\[
= U(\theta_0)^2/\mathcal{I}(\theta_0)
\]

As with the Wald statistic, the score statistic can be formulated in terms of the observed or expected information. Their asymptotic behaviour are identical, but for any finite sample size they may behave differently. Yet another variation of the score statistic is to use

\[
LR_{S} = 2[\ell(\hat{\theta}) - \ell(\theta_0)]
\]

which can be viewed as a compromise between the likelihood ratio and the score statistics. It happens that in the binomial situation \( \theta_0 + U(\theta_0)/\mathcal{I}(\theta_0) = \hat{\theta} \) for all \( \theta_0 \in (0,1) \), so \( LR_{S} \) is identical to the genuine score statistic, but that need not be the case in other situations.

In Figure 5.1 we have illustrated the log-likelihood function, the score approximation (green) and the Wald approximation (red) for the binomial situation with \( x = 8, n = 10 \) and \( \theta_0 = 0.45 \). The statistics are given as the difference in log-likelihood (LR) or relevant approximation there to (Wald and score) at the null and alternative hypotheses. The score statistic, \( S(\theta_0) = 4.949 \) is seen to be slightly smaller than the likelihood ratio statistic, \( LR(\theta_0) = 5.159 \), and Wald statistic \( W(\theta_0) = 7.656 \) somewhat larger then the other two.
Figure 5.1: Illustration of LR, score and Wald tests for the binomial situation with \( x = 8, n = 10 \) and \( H_0 : \theta_0 = 0.45 \). The log-likelihood is shown in black solid, the quadratic approximations related to the Wald and score statistics are shown in dashed red and dashed green respectively.

Figure 5.2: The score approximation (dashed blue) to the log-likelihood function (black solid) using the observed information.
5.2 Contrasting the likelihood ratio, score and Wald statistics via the shape of the likelihood function

5.2.1 Multiparameter settings

If the parameter of interest is a vector, say, $\theta$, the second order Taylor approximation to the log-likelihood function is

$$\tilde{\ell}(\theta, \theta_a) = \ell(\theta_a) + (\theta - \theta_a)^\top U(\theta_a) - \frac{1}{2} (\theta - \theta_a)^\top I(\theta_a) (\theta - \theta_a)$$

where

$$U(\theta) = \frac{\partial}{\partial \theta} \ell(\theta)$$

$$I(\theta) = -\frac{\partial^2}{\partial \theta \partial \theta^\top} \ell(\theta).$$

The (multivariate) Wald statistic reads

$$W(\theta_0) = (\theta_0 - \hat{\theta})^\top I(\hat{\theta})(\theta_0 - \hat{\theta})$$

and similarly the score statistic reads

$$S(\theta_0) = U(\theta_0)^\top I^{-1}(\theta_0) U(\theta_0),$$

Suppose we are only interested in a subset of the parameters. Let $\theta = (\theta^1, \theta^2)$, where $\theta^1 \in \mathbb{R}^q$, $\theta^2 \in \mathbb{R}^r$, $\theta \in \mathbb{R}^p$ and $p = q + r$. Let $\theta^2$ be the parameter of interest and we want to test $H_0 : \theta^2 = \theta^2_0$. Under $H_0$ we denote the parameter vector by $\theta_0 = (\theta^1_0, \theta^2_0)$ and let $\hat{\theta}_0 = (\hat{\theta}^1_0, \theta^2_0)$ denote the (conditional) ML estimate of $\theta$ given $\theta^2 = \theta^2_0$.

The likelihood ratio statistic for the test of this hypothesis reads

$$LR = 2\{\ell(\hat{\theta}) - \ell(\hat{\theta}_0)\}$$

$$= 2\{\ell(\hat{\theta}^1, \hat{\theta}^2) - \ell(\hat{\theta}^1_0, \theta^2_0)\}$$

$$= 2\{\ell(\hat{\theta}^1, \theta^2) - \ell_p(\theta^2_0)\}$$

where $\ell_p(\theta^2_0)$ is the profile log-likelihood of $\theta^2$ profiled over $\theta^1$ evaluated at $\theta^2_0$.

The profile log-likelihood is given by

$$\ell_p(\theta^2) = \ell_p\{\theta^2, \hat{\theta}^1(\theta^2)\} = \max_{\theta^1} \ell(\theta^1, \theta^2)$$

where the second formulation emphasizes that in general $\hat{\theta}^1$ is a function of $\theta^2$.

Equivalently we may write the likelihood ratio statistic as

$$LR = 2\{\ell_p(\theta^2) - \ell_p(\theta^2_0)\}$$

which motivates a relation between a quadratic approximation to $\ell_p(\theta^2)$ and the score and Wald statistics.
To develop the quadratic approximation to $\ell_p(\theta^2)$, consider the quadratic approximation to $\ell_p(\theta_0, \theta_1(\theta^2))$ as a function of $\theta^2$ where $\theta_1$ is expressed as a function of $\theta^2$ using (yet another) quadratic approximation to full log-likelihood, $\ell(\theta^1, \theta^2)$.

We use the following partitioning of the information matrix:

$$I = I(\theta) = I(\theta^1, \theta^2) = \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix}$$

and

$$I^{-1} = I^{-1}(\theta) = I^{-1}(\theta^1, \theta^2) = \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix}$$

note also that, e.g.

$$I^{22}(\theta) = (I_{22} - I_{21}I_{11}^{-1}I_{12})^{-1}$$

The second order Taylor approximation to the profile log-likelihood function is then

$$\tilde{\ell}_p(\theta^2, \theta_a^2) = \ell_p(\theta_a^2) + (\theta^2 - \theta_a^2)^	op U_p(\theta_a^2) - \frac{1}{2} (\theta^2 - \theta_a^2)^	op I_p(\theta_a^2)(\theta^2 - \theta_a^2)$$

The Wald type approximation to the likelihood ratio tests is

$$LR_W = 2\{\ell_p(\hat{\theta}^2) - \tilde{\ell}_p(\theta_a^2, \theta_1^2)\}$$

$$= 2\{\ell_p(\hat{\theta}^2) - \ell_p(\hat{\theta}_0^2) + \frac{1}{2} (\theta_0^2 - \hat{\theta}_0^2)^	op I_p(\hat{\theta}_0^2)(\theta_0^2 - \hat{\theta}_0^2)\}$$

$$= (\theta_0^2 - \hat{\theta}_0^2)^	op I_p(\hat{\theta}_0^2)(\theta_0^2 - \hat{\theta}_0^2).$$

The score statistic can be written in several ways; some are:

$$S(\theta_a^2) = U(\theta_0)^	op I(\theta_0)^{-1} U(\theta_0)$$

$$= U_p(\theta_0^2)^	op I^{22}(\theta_0) U(\theta_0^2)$$

$$= U_p(\theta_0^2)^	op I_p(\theta_0^2, \hat{\theta}_0^2)^{-1} U(\theta_0^2)$$

$$= U_p(\theta_0^2)^	op \{I_{22} - I_{21}I_{11}^{-1}I_{12}\} U(\theta_0^2).$$

The score statistic can be derived from the quadratic approximation to the profile likelihood as follows:

$$LR_S = 2\{\tilde{\ell}_p(\theta_a^2, \theta_0^2) - \ell_p(\theta_0^2)\}$$

$$= 2\{\ell_p(\theta_0^2) + (\theta_0^2 - \theta_0^2) U_p(\theta_0^2) - \frac{1}{2} (\theta_0^2 - \theta_0^2)^	op I_p(\theta_0^2)(\theta_0^2 - \theta_0^2) - \ell_p(\theta_0^2)\}$$

$$= 2(\theta_0^2 - \theta_0^2)U_p(\theta_0^2) - (\theta_0^2 - \theta_0^2)^	op I_p(\theta_0^2)(\theta_0^2 - \theta_0^2)$$

$$= U_p(\theta_0^2)^	op I_p(\theta_0^2, \hat{\theta}_0^2)^{-1} U(\theta_0^2).$$
5.3 Explaining the Hauck-Donner effect via the shape of the likelihood function

Hauck Jr. and Donner (1977) showed that the Wald statistic of a regression parameter in a logistic regression model decreases to zero as the distance between the parameter estimate and the value under the null increases. They also showed that the power of the Wald test decreases to the significance level for alternatives sufficiently far from the null. They showed this using rather technical asymptotic
arguments. Here we explain their results in terms of the shape of the log-likelihood function and extend their comparison of Wald and LR tests with the score test.

The setting is this: The hypothesis test of interest is the test of a group contrast \((H_0: \beta = 0)\) where the proportion of successes in group 1 is \(p_1 = 0.25\) and \(p_2\) in group 2. There are \(n = 100\) observations in each group and the logistic regression model contains an intercept (baseline for group 1 say) in addition to the group contrast parameter.

In Figure 5.3 we have taken \(p_2 = 0.99\). The figure shows the log-likelihood function (black solid) as a function of the group contrast parameter, the quadratic approximation to this at the MLE, \(\hat{\beta}_1\) (dashed black) related to the Wald statistic and the quadratic approximation at the null (dotted blue) related to the score statistic. As is obvious from the figure, the log-likelihood function is extremely flat around the MLE and beyond \(\beta_1 = 5\) and this causes an extreme discrepancy between LR and Wald statistics. Had the log-likelihood function been closer to a quadratic function, all three (LR, Wald and score) statistics would agree.

The LR statistic is twice the difference in log-likelihood at the alternative (optimum) and null values. The Wald statistic is essentially the same statistic, but where the value under the null is approximated by the dashed curve, which clearly leads to a much smaller statistic. The score test is the difference between the value of the dotted curve under the null (which coincides with the solid black curve by definition) and the value at the stationary point. This is seen to provide a much better approximation to the LR statistic than the Wald statistic although it is still a little smaller than the LR statistic.

Note that the shape of the log-likelihood function is (in part) determined by the scale at which the parameter is considered. On the probit scale the discrepancy between the three statistics appear to occur for more extreme \(p_2\) than on the logit scale.

Hauck Jr. and Donner (1977) include a comparison of Wald and LR statistics for a range of \(p_2\). In the following table we have replicated this (part 2 of their table) and added the score test for comparison. The score test is seen to behave similar to the LR test although it also appears to be less powerful than the LR test (smaller magnitude of the test statistic).

<table>
<thead>
<tr>
<th>beta</th>
<th>Wald</th>
<th>LR</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>-3.497</td>
<td>11.496</td>
<td>30.887</td>
</tr>
<tr>
<td>0.02</td>
<td>-2.793</td>
<td>13.845</td>
<td>26.238</td>
</tr>
<tr>
<td>0.03</td>
<td>-2.377</td>
<td>14.239</td>
<td>22.570</td>
</tr>
<tr>
<td>0.04</td>
<td>-2.079</td>
<td>13.782</td>
<td>19.519</td>
</tr>
<tr>
<td>0.05</td>
<td>-1.846</td>
<td>12.912</td>
<td>16.914</td>
</tr>
<tr>
<td>0.1</td>
<td>-1.099</td>
<td>7.340</td>
<td>8.007</td>
</tr>
<tr>
<td>0.15</td>
<td>-0.636</td>
<td>3.070</td>
<td>3.152</td>
</tr>
<tr>
<td>0.35</td>
<td>0.480</td>
<td>2.364</td>
<td>2.389</td>
</tr>
</tbody>
</table>
5.3 Explaining the Hauck-Donner effect via the shape of the likelihood function

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.45</td>
<td>0.898</td>
<td>8.602</td>
<td>8.884</td>
<td>8.791</td>
</tr>
<tr>
<td>0.55</td>
<td>1.299</td>
<td>18.009</td>
<td>19.110</td>
<td>18.750</td>
</tr>
<tr>
<td>0.65</td>
<td>1.718</td>
<td>30.325</td>
<td>33.299</td>
<td>32.323</td>
</tr>
<tr>
<td>0.75</td>
<td>2.197</td>
<td>45.261</td>
<td>52.325</td>
<td>50.000</td>
</tr>
<tr>
<td>0.85</td>
<td>2.833</td>
<td>60.920</td>
<td>78.247</td>
<td>72.727</td>
</tr>
<tr>
<td>0.9</td>
<td>3.296</td>
<td>66.056</td>
<td>95.258</td>
<td>86.445</td>
</tr>
<tr>
<td>0.91</td>
<td>3.412</td>
<td>66.370</td>
<td>99.142</td>
<td>89.409</td>
</tr>
<tr>
<td>0.92</td>
<td>3.541</td>
<td>66.270</td>
<td>103.230</td>
<td>92.452</td>
</tr>
<tr>
<td>0.95</td>
<td>4.043</td>
<td>61.951</td>
<td>117.035</td>
<td>102.083</td>
</tr>
<tr>
<td>0.99</td>
<td>5.694</td>
<td>30.485</td>
<td>141.958</td>
<td>116.214</td>
</tr>
</tbody>
</table>

An interesting question is how much value can be put to statements such as “the Wald, score and LR tests are asymptotically equivalent” when their behavior can differ so dramatically.

Given that Wald and score statistics are only approximations to the LR statistic one could argue that the LR statistic is always preferred. Indeed the LR statistic seem the most honest, or best, measure of support or evidence of a hypothesis over another. The LR statistic is parameterization invariant - a characteristic not shared by the Wald statistics, so at any particular scale it may, or may not, closely/badly approximate the LR statistic. However, when a \( p \)-value is to be found, the score (or Wald) statistic may more closely follow a \( \chi^2 \)-distribution under the null in some particular situation at some particular parameterization.


Richardson, L. F. (1910). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 210, 307–357.


Appendix A

Estimation and Inference in the Same-Different test

Estimation and inference in the same–different test

Rune Haubo Bojesen Christensen*, Per Bruun Brockhoff

DTU Informatics, Technical University of Denmark, Richard Petersens Plads, Building 321, DK-2800 Kongens Lyngby, Denmark

Abstract

Inference for the Thurstonian $\delta$ in the same–different protocol via the well known Wald statistic is shown to be inappropriate in a wide range of situations. We introduce the likelihood root statistic as an alternative to the Wald statistic to produce CIs and $p$-values for assessing difference as well as similarity. We show that the likelihood root statistic is equivalent to the well known $G^2$ likelihood ratio statistic for tests of no difference. As an additional practical tool, we introduce the profile likelihood curve to provide a convenient graphical summary of the information in the data about $\delta$.

On the basis of simulations, we show that the coverage probability of the Wald-based 95% CI for $\delta$ is often far from 95%, whereas the coverage probability of the profile likelihood-based CI is close to the desired 95%. We also show how the likelihood framework can be used to combine information from independent experiments possibly using different discrimination protocols to obtain inference for a common $\delta$. Finally, we provide a free R package with an implementation of the likelihood methodology presented in this paper.

Keywords:
Profile likelihood curve
Thurstonian models
Discrimination protocols
Similarity

1. Introduction

Miss Anna Sens, a sensory analyst, uses the ordinary two-interval same–different test to examine the sensory difference between two products via $\delta$, the Thurstonian measure of sensory difference. She obtains the data in Table 1. The estimate of $\delta$ is $\hat{\delta} = 1.88$ and a 95% normal-based (Wald) confidence interval (CI) is $[0.51, 3.26]$. The corresponding one-sided $p$-value is $p = 0.00369$ for the hypothesis of no difference: $H_0 : \delta = 0$ versus $H_1 : \delta > 0$. Anna now believes that the difference between the products is so large that assessors were able to discriminate between them reliably. In this paper we show that the estimate of $\delta$ is in fact the maximum likelihood estimate (MLE). We also show that the 95% profile likelihood CI for $\delta$, $[0.00, 3.21]$ and the corresponding one-sided likelihood-based $p$-value of 0.0563 are more accurate. The likelihood-based $p$-value is markedly different from the inaccurate Wald $p$-value and shows far less evidence against the null hypothesis.

Mr. Mark Ting also has two products, and with the same–different protocol, he obtains the data in Table 2. He is also interested in estimating the sensory difference between the products, but in contrast to Anna, his ultimate goal is similarity. He decides that the products are sufficiently similar if the difference ($\delta$) is 1.5 or smaller. The fraction of “same” answers to same samples (8/13) is smaller than “same” answers to different samples (11/13) in Mark’s data, so the only sensible estimate of $\delta$ is 0 which is also the MLE. Mark is unable to construct a CI with any previously published method. We show that a 95% profile likelihood-based CI for $\delta$ is $[0.00, 1.34]$. Mark can be reasonably sure that the products are sufficiently similar for his purposes because the upper confidence limit is below 1.5.

The first example above illustrates that inference based on the traditional Wald statistic can be misleading. The second example illustrates the need for statistical methods when the Wald statistic is not applicable. Although the same–different test is a well known test in the sensometric literature, only little literature is concerned with inference and statistical analyses from same–different tests. Data from a same–different test are conveniently summarized in a $2 \times 2$ table which suggests that conventional statistical methods for such tables can be used. To assess the null hypothesis of no sensory difference, Bi (2006) suggested Pearson’s $\chi^2$ statistic, $X^2$ and Fisher’s exact test. An alternative is the likelihood ratio statistic statistic $G^2$.

The $X^2$, $G^2$ and exact tests are non-parametric in the sense that they do not require the analyst to adopt a particular parameterization of the same–different test. However, they only apply when the null hypothesis is that of no sensory difference. If other null hypotheses are of interest, for instance hypotheses of similarity, a measure of sensory difference, i.e. a parameterization of the same–different test, has to be chosen.

Macmillan, Kaplan, and Creelman (1977) derived the Thurstonian parameterization of the same–different test assuming a differing strategy (or $\tau$ strategy), where the Thurstonian $\delta$ is the measure of sensory difference. The differing strategy seems to be the most widely assumed strategy for the same–different test in the sensory literature; see O’Mahony and Rousseau (2002) for an overview and further references. Bi (2002) derived the variance...
of $\delta$ for the parameterization developed by Macmillan et al. (1977). Although Bi (2002) did not discuss how statistical inference can be obtained using the variance of $\delta$, it seems natural to use the well known Wald statistic to obtain CIs and $p$-values for hypotheses of difference and similarity. We show, however, in this paper that the Wald statistic is frequently inappropriate because it gives misleading inference in some cases and is not even computable in other cases.

The purpose of this paper is to present a statistical methodology that provides accurate inference for the Thurstonian parameterization of the same–different test assuming a differencing strategy and to show how the theory of likelihood provides a convenient framework for this.

The likelihood function is a powerful statistical tool that extracts the available information from the data. This paper illustrates how the theory of likelihood can be used to extract more information from discrimination experiments than that provided by tests of standard hypotheses.

In the following sections, we show how likelihood procedures provide a unified framework for estimation of parameters, CIs and $p$-values for hypotheses of difference and similarity. We also provide the free R package sensR (Christensen & Brockhoff, 2008) for R (R Development Core Team, 2008) that facilitates application of the developed methodology.

The same–different protocol has been advocated because of its high sensitivity observed in experiments compared to alternatives such as the triangle and duo-trio tests (Rousseau & O'Mahony, 2001; Rousseau, Meyer, & O'Mahony, 1998; Stillmann & Irwin, 1995). The high sensitivity is believed to be due to its suitable cognitive strategy and low memory requirements (Rousseau & O'Mahony, 2000; Rousseau, Rogeaux, & O'Mahony, 1999). For these reasons, and because the test is conceptually simple and easily understood, the test is often advocated for studies using untrained assessors as is often the case in consumer studies (e.g. O'Mahony & Rousseau, 2002). An extension of the same–different protocol to allow answers on a rating scale has been proposed (Irwin, Stillman, Hautus, & Huddleston, 1993). Likelihood-based analysis is also possible for this model, but is not the topic of the current paper.

The methods we discuss apply to both the short version and under a mild assumption of independent observations also to the long version of the test. In the short version each individual receives either a same or a different pair. In the long version each individual receives both a same and a different pair, unaware that this is the case (see e.g. O'Mahony & Rousseau, 2002). We follow general statistical practice and add a ‘hat’ to the estimate of a parameter, e.g. $\hat{\delta}$, but note that $\hat{d}$ ($\hat{d}$-prime) is elsewhere used instead of $\hat{\delta}$. The methods we propose are based on the assumptions that all observations are independent and that the distributions of sensory intensity have the same variance and are uncorrelated, i.e. that Thurstone’s case V (Thurstone, 1927) applies.

This paper proceeds as follows. In Section 2 we discuss the likelihood function for the same–different model and show that previously proposed estimators of $\tau$ and $\delta$ coincide with ML estimators in well behaved situations. In Section 3 we introduce the profile likelihood curve, we discuss how the Wald and likelihood root statistics can be used to obtain CIs, and we discuss statistics and $p$-values for hypotheses for assessing difference and similarity. We also show how inference for a common $\delta$ can be obtained in the likelihood framework when information is available from several experiments using different discrimination protocols. In Section 4 we consider the distribution of the ML estimator of $\delta$ and assess the coverage probability of likelihood and Wald–based CIs. This provides a basis for an assessment of the appropriateness of the Wald and likelihood root statistics. In Section 5 we revisit the examples of the introduction to illustrate the methodology developed. We end with a discussion in Section 6 and conclusions in Section 7.

Table 1

<table>
<thead>
<tr>
<th>Sample</th>
<th>Answer</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“same”</td>
<td>“different”</td>
</tr>
<tr>
<td>Same</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Different</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Sample</th>
<th>Answer</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“same”</td>
<td>“different”</td>
</tr>
<tr>
<td>Same</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Different</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>19</td>
<td>7</td>
</tr>
</tbody>
</table>


2. The likelihood function and ML estimators

The same–different test involves two stimuli variables (e.g. variations of a product), say X and Y. The Thurstonian law (Thurstone, 1927) assumes that the perceptions elicited by X and Y can be described by two normal distributions. In the development of the test, Macmillan et al. (1977) further assumed that Thurstone’s case V applies, i.e. that the distributions have equal variance, are uncorrelated and may differ only in their location, corresponding to the standard assumptions in signal detection theory. That difference in location is the Thurstonian $\delta$ – our parameter of interest. The same–different test involves one more parameter, a decision criterion, $\tau$ which determines how large the difference between stimuli has to be in order to be judged different. If the perceptual difference between two stimuli is less than $\tau$, they are judged “same” and if the difference is larger than $\tau$, they are judged “different”. The parameters, $\tau$ and $\delta$ are scaled with the standard deviation of the normal distributions, i.e. the perceptual standard deviation.

The observations from a same–different test are distributed as the product of two binomial distributions, and the data can fall in one of four distinct classes. The answer can be “same” or “different” when the pair of observations are either same of different. Let

$$p_{ss} = 2\Phi(\tau / \sqrt{2}) - 1; \quad p_{sd} = \Phi(\tau - \delta / \sqrt{2}) - \Phi(\tau - \delta / \sqrt{2})$$

where $\Phi$ is the standard normal CDF, $p_{ss} = 1 - p_{sd}$ and $p_{sd} = 1 - p_{ss}$. Assuming independent data, the log-likelihood function for the same–different protocol is given by

$$l(\tau; \delta; y) = \log \frac{f_{bin}(p_{ss}; n_{s}, n_{d}) f_{bin}(p_{sd}; n_{s}, n_{d})}{c + \sum n_{j} \log p_{j}}$$

(1)

where $n_{s}$ and $n_{d}$ are the numbers of same samples and different samples respectively, $n_{j}$ is the number of observations in the $j$th category, $c$ is a constant with respect to $\tau$ and $\delta$, and $f_{bin}(p; m)$ is the binomial density with success probability $p$ and index $m$. 


Table 1

<table>
<thead>
<tr>
<th>Sample</th>
<th>Answer</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“same”</td>
<td>“different”</td>
</tr>
<tr>
<td>Same</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Different</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Sample</th>
<th>Answer</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“same”</td>
<td>“different”</td>
</tr>
<tr>
<td>Same</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Different</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>19</td>
<td>7</td>
</tr>
</tbody>
</table>
Provided data are obtained in all four classes and the fraction of "same" answers to same samples is larger than the fraction of "same" answers to different samples, the MLEs of $\tau$ and $\delta$ are given as solutions to (Appendix B)

\[
\hat{\tau} = \sqrt{2} \Phi^{-1} \left( \frac{2n_{12} + n_{66}}{2(n_{16} + n_{61})} \right) \tag{2}
\]

\[
\hat{\delta} = \frac{n_{16} + n_{61}}{n_{16} + n_{61}} \Phi \left( \frac{\hat{\tau} - \hat{\delta}}{\sqrt{2}} \right) - \Phi \left( \frac{-\hat{\tau} - \hat{\delta}}{\sqrt{2}} \right) \tag{3}
\]

where the estimate of $\delta$ has to be obtained by numerical methods. These estimators are identical to those derived by Macmillan et al. (1977), Kaplan, Macmillan, and Creelman (1978) and also reported by O’Mahony and Rousseau (2002), but to our knowledge they have not previously been justified as ML estimators. The identification of the estimators as ML estimators is important because ML estimators are known to possess good properties. They are for instance consistent and asymptotically optimal (see e.g. Wasserman, 2004), so they use all the information in the data.

When data are not obtained in all four classes or the fraction of “same” answers to same samples is not larger than the fraction of “same” answers to different samples, the estimates of $\tau$ and $\delta$ cannot be computed from (2) and (3), but in several cases MLEs exist nonetheless. We have summarized the 6 possible cases in Table 3. Note that there is no information about $\delta$ when no “same” answers or no “different” answers are obtained (cases V and VI, respectively). In all other cases inference for $\delta$ is possible.

### 3. Profile likelihood and measures of evidence

In this section we discuss how inference can be obtained in the same–different test. We introduce the likelihood root statistic as an alternative to the well known Wald statistic and show that the latter is an approximation to the former. We introduce the profile likelihood curve of $\delta$, a central and fundamental entity. We show how the likelihood root statistic reduces to the likelihood ratio statistic, which is given as $LR(\hat{\delta}_0) = r^2(\hat{\delta})$ and follows a $\chi^2$-distribution under the null hypothesis. A related statistic is the Wald statistic, $t(\hat{\delta}, \hat{\delta}_0)$, in that the Wald statistic corresponds to the likelihood root statistic if the profile likelihood is approximated by a symmetric Gaussian curve (Pawitan, 2001). The approximation improves if the sample size increases and in general also if $\hat{\delta}$ increases but stays finite.

The variance of $\hat{\delta}$ is a critical ingredient in the Wald statistic. The variance of $\hat{\delta}$ can be defined as the appropriate diagonal entry of the inverse of the observed Fisher information matrix (Efron & Hinkley, 1978; Pawitan, 2001), i.e. the negative Hessian of the log-likelihood function evaluated at the MLE (see Appendix C). Bi (2002) derived the variance of $\hat{\delta}$ using the delta method (see e.g. Wasserman, 2004) which leads to the same result.

The variance of $\hat{\delta}$ is only finite for positive and finite $\hat{\delta}$ (see Appendix C) which only happens when data fall in case I (c.f. Table 3). We will refer to the variance as well as the Wald statistic as "computable" when data fall in case I.

The profile likelihood for $\delta$ cannot be symmetric, not even approximately, in other than case I. This is because the MLE of $\delta$ is at the boundary of the parameter space (at either zero or infinity) in cases II–IV, so the likelihood function is monotonically increasing or decreasing as a function of $\delta$ and therefore not symmetric. Even if a finite variance estimate could be found in cases II–IV, the Wald statistic would therefore still be inappropriate in these cases.

Next, we discuss three tools to achieve statistical inference: The profile likelihood curve, the CI and the p-value.

Traditionally the Wald statistic

\[
w(\hat{\delta}_0) = \frac{\hat{\delta} - \delta_0}{\sigma(\hat{\delta})}
\]

is used to obtain CIs and p-values. We propose an alternative statistic: the likelihood root statistic (see e.g. Brazzale, Davison, & Reid, 2007; Severini, 2000)

\[
r(\hat{\delta}_0) = \text{sign}(\hat{\delta} - \delta_0) \sqrt{2 \left( l_c(\hat{\delta}_0, y) - l_c(\delta_0, y) \right)^{1/2}}
\]

They both asymptotically follow a standard normal distribution under the null hypothesis. A related statistic is the Wald statistic, $t(\hat{\delta}_0)$, in that the Wald statistic corresponds to the likelihood root statistic if the profile likelihood is approximated by a symmetric Gaussian curve (Pawitan, 2001). The approximation improves if the sample size increases and in general also if $\hat{\delta}$ increases but stays finite.

The variance of $\hat{\delta}$ is a critical ingredient in the Wald statistic. The variance of $\hat{\delta}$ can be defined as the appropriate diagonal entry of the inverse of the observed Fisher information matrix (Efron & Hinkley, 1978; Pawitan, 2001), i.e. the negative Hessian of the log-likelihood function evaluated at the MLE (see Appendix C). Bi (2002) derived the variance of $\hat{\delta}$ using the delta method (see e.g. Wasserman, 2004) which leads to the same result.

The variance of $\hat{\delta}$ is only finite for positive and finite $\hat{\delta}$ (see Appendix C) which only happens when data fall in case I (c.f. Table 3). We will refer to the variance as well as the Wald statistic as "computable" when data fall in case I.

The profile likelihood for $\delta$ cannot be symmetric, not even approximately, in other than case I. This is because the MLE of $\delta$ is at the boundary of the parameter space (at either zero or infinity) in cases II–IV, so the likelihood function is monotonically increasing or decreasing as a function of $\delta$ and therefore not symmetric. Even if a finite variance estimate could be found in cases II–IV, the Wald statistic would therefore still be inappropriate in these cases.

Next, we discuss three tools to achieve statistical inference: The profile likelihood curve, the CI and the p-value.

**Table 3**

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>$\tau$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$n_{12}/(n_{12} + n_{60}) &gt; n_{16}/(n_{16} + n_{61})$</td>
<td>$\hat{\tau}$</td>
<td>$\hat{\delta}$</td>
</tr>
<tr>
<td>II</td>
<td>$n_{12}/(n_{12} + n_{60}) \leq n_{16}/(n_{16} + n_{61})$ or $n_{12} = 0$</td>
<td>$\hat{\tau}$</td>
<td>0</td>
</tr>
<tr>
<td>III</td>
<td>$n_{16} = 0$ or $n_{16} = n_{12} = n_{60}$</td>
<td>$\hat{\tau}$</td>
<td>$\hat{\delta}$</td>
</tr>
<tr>
<td>IV</td>
<td>$n_{12} = 0$ or $n_{12} = n_{16} = n_{60}$</td>
<td>0</td>
<td>$\infty$</td>
</tr>
<tr>
<td>V</td>
<td>$n_{12} = n_{16} = 0$</td>
<td>0</td>
<td>NA</td>
</tr>
<tr>
<td>VI</td>
<td>$n_{12} = n_{16} = 0$</td>
<td>$\infty$</td>
<td>NA</td>
</tr>
</tbody>
</table>

Notation: Hat-notation indicates that the MLE is positive, finite and depends further on the data, NA stands for “Not Available”, i.e. no information on the parameter.

**Fig. 1.** Profile likelihood (solid) of $\delta$ for Anna’s experiment. Horizontal lines denote 95% and 99% confidence limits. Dashed line is the symmetric Gaussian approximation.
3.1. Profile likelihood curve

The normalized profile likelihood measures the relative plausibility or likelihood of hypothesized values of \( \delta \) relative to \( \delta \). The profile likelihood curve for \( \delta \) is a convenient graphical description of the information about \( \delta \) in the data. An example of such a curve is given in Fig. 1 and may help fix ideas. The figure will be thoroughly discussed in Section 5. The normalized profile likelihood for \( \delta \) is given by

\[
\mathcal{L}_{\delta}(\delta; y) = \exp \left\{ I_{\delta}(\delta; y) - I_{\delta}(\hat{\delta}; y) \right\}
\]

and the profile likelihood curve for \( \delta \) is a plot of \( \mathcal{L}_{\delta}(\delta; y) \) versus \( \delta \). \( \mathcal{L}_{\delta}(\delta; y) \) is bounded between zero and one and reaches the upper bound at the MLE, \( I_{\delta}(\hat{\delta}; y) \). The profile likelihood curve is a fundamental property: the likelihood root and likelihood ratio statistics are simple transformations of \( \mathcal{L}_{\delta}(\delta; y) \) for a particular value of \( \delta \). CIs and \( p \)-values are also functions of the normalized profile likelihood, \( \mathcal{L}_{\delta}(\delta; y) \).

The Wald approximation to \( \mathcal{L}_{\delta}(\delta; y) \) is given by the symmetric Gaussian curve \( \mathcal{W}(\delta; y) = \exp(-\frac{1}{2} \omega(\delta)^2) \).

3.2. Confidence intervals

A 100(1 − \( \alpha \)) per cent CI for \( \delta \) is given by those values of \( \delta \) that satisfy

\[
\mathcal{C} : I_{\delta}(\delta; y) < Z_{1-\alpha/2}
\]

where \( I(\hat{\delta}) \) is the Wald or likelihood root statistic and \( Z_{1-\alpha/2} \) is the 1 − \( \alpha /2 \) quantile of the standard normal distribution.

The Wald CI is close to the profile likelihood CI when the profile likelihood of \( \delta \) is well approximated by a symmetric Gaussian curve.

For a CI to be informative about \( \delta \) in a particular experiment it is required that the CI contains those values of the parameter most supported by the data in the sense that no values outside the interval should be more supported by the data than values inside the interval. This principle is known as the law of likelihood (c.f. Boyles, 2008; Hacking, 1965; Royall, 1997). CIs based on the profile likelihood conform with this property. The Wald CIs conform with this principle approximately when the profile likelihood of \( \delta \) is well approximated by a symmetric Gaussian curve.

The height of the normalized profile likelihood curve is uniquely related to a CI at a given \( \alpha \), so intersections between the profile likelihood curve, \( \mathcal{L}_{\delta}(\delta; y) \) and horizontal lines define CIs. It can be shown (e.g. Pawitan, 2001) that the horizontal lines are at \( \exp(-\frac{1}{2} \Omega_{1-\alpha/2}^2) \) so for 95% and 99% CIs the lines are at 0.1465 and 0.03625 respectively (see Fig. 1 for an illustration). The bounds of CIs based on the Wald approximation are given similarly as the intersections between the horizontal lines and the Wald approximation to the normalized profile likelihood, \( \mathcal{W}(\delta; y) \). The profile likelihood curve is a particularly helpful interpretational device when it is augmented with cutoffs indicating CIs at a few relevant levels such as the 95% and 99% levels.

We can view the CI and MLE as a three-value summary of the profile likelihood curve subject to a specific choice of \( \alpha \). Conversely, we may view the profile likelihood curve as the collection of CIs for all values of \( \alpha \).

3.3. \( p \)-Values for hypotheses of difference and similarity

The Wald and likelihood root statistics can be used to obtain \( p \)-values so as to measure the support for one hypothesis over another. Suppose the following hypotheses for assessing a sensory difference:

\[
H_0 : \delta \leq \delta_0 \quad \text{versus} \quad H_1 : \delta > \delta_0
\]

The \( p \)-value is \( p = 1 - \Phi(t(:\delta_0)) \), where \( t(\cdot) \) represents the Wald or the likelihood root statistic.

When the null hypothesis is that of zero association, the likelihood ratio statistic, \( LR(\delta_0) \) reduces to the well known \( G^2 \) statistic (e.g. Agresti, 2002)

\[
LR(0) = G^2 = 2 \sum_{j=1}^{n} \frac{\log n_j/e_j}{e_j}
\]

where \( n_j \) is the observed frequency, and \( e_j \) is the expected frequency under the null hypothesis in the \( j \)th cell. Since \( \delta \) is non-negative, the test is one-sided, so the \( p \)-value is identical to that reported by the likelihood root statistic and is given as \( p = (1 - F(G^2))/2 \), where \( F \) is the CDF of a \( G^2 \) variable with one degree of freedom.

The \( G^2 \) statistic is asymptotically identical to Pearson's \( X^2 \) statistic

\[
X^2 = \sum_{j} \frac{(n_j - e_j)^2}{e_j}
\]

Notice that it is Pearson's \( X^2 \) statistic without Yates' continuity correction that is considered here. Fisher's exact test has elsewhere (ASTM, 2005; Bi, 2006) been suggested for analysis of same–different experiments. This test, however, assumes a multinomial sampling situation with fixed margins rather than the product binomial in the same–different protocol. Berkson (1978) showed that Pearson's \( G^2 \) test with Yates' continuity correction is an approximation to Fisher's exact test (see also Yates, 1984). Berkson (1978) also showed that Fisher's exact test and Pearson's \( G^2 \) test with Yates' continuity correction are conservative under the product binomial sampling situation and that the uncorrected Pearson's \( G^2 \) statistic leads to rejection rates of the desired size. Fisher's exact test and Pearson's \( X^2 \) statistic with Yates' continuity correction are therefore inaccurate for same–different experiments. For further discussion and additional viewpoints, see also Yates (1984) and Little (1989).

Although the \( G^2 \) and \( X^2 \) statistics are asymptotically equivalent, they may perform quite differently under various patterns of sparseness and zeros in the data. However, they are often equivalent in well behaved cases (Agresti, 2002; Cressie & Read, 1989). The \( G^2 \) approximation will in general be acceptable for both tests if the expected frequencies are five or more in all cells.

The traditional choice of \( \delta_0 \) is zero when a \( p \)-value is sought for the difference hypotheses, so the null hypothesis corresponds to that of no sensory difference or no association. However, it is often insightful to consider a value of \( \delta_0 \) that corresponds to a hypothesis of an ignorable versus a relevant sensory difference.

Notice that when \( \delta_0 \) is chosen different from zero, the \( G^2 \) and \( X^2 \) statistics are inapplicable, whereas the Wald and likelihood root statistics are applicable.

When a researcher is interested in the evidence of similarity, a \( p \)-value can be found for the hypotheses

\[
H_0 : \delta \geq \delta_0 \quad \text{versus} \quad H_1 : \delta < \delta_0
\]

There is no standard choice of \( \delta_0 \) for the similarity hypothesis, but a natural choice is the smallest \( \delta_0 \) of practical or scientific relevance. \( p \)-values can be obtained from the likelihood root and Wald statistics and are again given as \( p = 1 - \Phi(t(:\delta_0)) \), where \( t(\cdot) \) represents the chosen statistic.

The assessment of similarity has been discussed several times in the sensory literature via so-called equivalence tests (Bi, 2005, 2007, 2008; Meyners, 2007, 2008; Ennis, 2007, 2008). They have has also been the subject of heated discussion in the statistical literature (Berger & Hsu, 1996; Perlman & Wu, 1999; Schuirmann, 1981, 1987). When assessing similarity via \( \delta \) in a discrimination protocol, the controversial issue does not arise since the alternative hypothesis is one-sided.
Notice that to obtain a p-value, a choice of an appropriate set of hypotheses has to be made, while to obtain a CI, only a choice of \( z \) has to be chosen. The same CI can therefore be used to assess difference as well as similarity, as was also recommended by Carr (1995). The profile likelihood curve can be used similarly and lightens the problem of choosing an appropriate \( z \). Many authors have argued that CIs are often of higher scientific and practical relevance than p-values because they reflect the information in the data irrespective of any chosen set of hypotheses (Blume & Peiperl, 2003; Cox, 2006; Cox & Hinkle, 1974; Nester, 1996; Goodman, 1999).

### 3.4. Choice of parameterization in the same–different protocol

The \( X^2 \) and \( G^2 \) statistics do not require any parameterization or choice of measure of sensory difference when they are used to assess the hypothesis of no sensory difference. When difference hypotheses where the null effect is larger than zero or similarity hypotheses are contemplated, a particular parameterization, that is a measure of effect size, has to be chosen.

Bi (2005) proposed to use the test of Dunnett and Gent (1977) to assess the similarity of sensory differences in the same–different protocol and consequently use the difference, \( p_{ss} - p_{dd} \), i.e. the difference in the probability of a “same” answer between same samples and different samples, as a measure of sensory difference. Because the difference, \( p_{ss} - p_{dd} \), is not a unique transformation of \( \delta \), the two parameterizations lead to different conclusions.

To illustrate this, consider the following two experiments, each with 100 same samples and 100 different samples:

**A:** \( p_{ss} = 0.5 \) and \( p_{dd} = 0.1 \)

**B:** \( p_{ss} = 0.9 \) and \( p_{dd} = 0.5 \)

In both experiments \( p_{ss} - p_{dd} = 0.4 \), but \( \delta_A = 2.73 \) and \( \delta_B = 2.32 \), so using the difference, \( p_{ss} - p_{dd} \), as a measure of sensory difference leads to the same inference, whereas using the Thurstonian \( \delta \) as a measure of sensory difference does not lead to the same inference in the two experiments. Since \( \tau \) is a one-to-one transformation of \( \delta \), given the sample size (see (2)) it also differs and the MLEs are \( \tau_A = 0.954 \) and \( \tau_B = 2.33 \).

Bi and Ennis (2001) similarly found that the difference in the probability of an “A” answer in a –Not A protocol between A and Not A samples is not uniquely related to the Thurstonian measure of sensory difference, \( \delta \).

Notice that the parameterization problem seldom arises in the discrimination protocols based on a single binomial experiment and defined by their psychometric functions such as the triangle, duo-trio and m-AFC protocols. In these protocols, the probability of a correct answer, \( P \) is uniquely linked to the Thurstonian \( \delta \) via the psychometric function. Because the information in the data about \( \delta \) is identical to that about \( P \), proper statistical analyses lead to identical conclusions irrespective of the parameterization.

### 3.5. Combining information from experiments using different discrimination protocols

If several independent experiments, possibly using different protocols, provide information on a common parameter, \( \delta \) then the likelihood function can be used to obtain inference for \( \delta \). Consider two experiments with probability models \( f_1(\delta, \psi_1; y_1) \) and \( f_2(\delta, \psi_2; y_2) \), where \( \psi_1 \) and \( \psi_2 \) are potential nuisance parameters such as \( \tau \) in the same–different protocol. The profile log-likelihood for \( \delta \) is

\[
\ell_{\psi_1, \psi_2}(\delta; y_1, y_2) = \arg \max \left\{ \ell_1(\delta, \psi_1) + \ell_2(\delta, \psi_2) \right\}
\]

Confidence intervals and p-values are obtained via likelihood root or Wald statistics as before.

The concept extends to an arbitrary number of independent experiments. The asymptotic theory is, however, only appropriate when the number of nuisance parameters is not large and, notably, does not increase with the sample size (e.g. Pawitan, 2001).

We might hypothesize that \( \delta \) actually takes on two different values, \( \delta_1 \) and \( \delta_2 \) in the two experiments. We can use the likelihood ratio statistic

\[
LR = 2 \left\{ \ell_{\psi_1, \psi_2}(\delta_1, \psi_1; y_1, y_2) - \ell_{\psi_1, \psi_2}(\delta_2, \psi_2; y_1, y_2) \right\}
\]

and corresponding p-value to measure the evidence against that hypothesis. The distribution of LR is asymptotically that of a \( \chi^2 \) variable under the null hypothesis with degrees of freedom equal to the difference in parameters of the two likelihoods – in this particular case one degree of freedom.

### 4. Distribution of the ML estimator of \( \delta \) and coverage probability of likelihood and Wald CIs

In this section we examine how the ML estimator of \( \delta \) distributes in the 6 classes summarized in Table 3. We also examine the coverage probability of the Wald and profile likelihood CIs. The distribution of the ML estimator of \( \delta \) can be derived from binomial probabilities as briefly outlined in Appendix D. The coverage probabilities are estimated via simulations. The coverage

<table>
<thead>
<tr>
<th>N</th>
<th>( \tau )</th>
<th>Case</th>
<th>( \delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0.5</td>
<td>I</td>
<td>0.46 0.49 0.57 0.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.53 0.51 0.43 0.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>0.47 0.51 0.62 0.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.53 0.49 0.38 0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>100</td>
<td>0.5</td>
<td>I</td>
<td>0.46 0.53 0.57 0.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.54 0.53 0.47 0.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>0.46 0.49 0.57 0.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.54 0.51 0.43 0.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>20</td>
<td>0.5</td>
<td>I</td>
<td>0.45 0.48 0.58 0.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.61 0.60 0.58 0.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0.04 0.04 0.05 0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>0.41 0.42 0.46 0.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.59 0.58 0.54 0.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.23 0.24 0.28 0.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>II</td>
<td>0.59 0.58 0.54 0.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>III</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IV</td>
<td>0.15 0.15 0.16 0.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VI</td>
<td>0.03 0.03 0.02 0.01</td>
</tr>
</tbody>
</table>

* Number of same and different samples each equal to \( N/2 \).
probability of a CI-producing rule (such as profile likelihood or Wald) is the probability that a CI produced by the rule will contain the true parameter.

We have summarized the distribution of the ML estimator of \( \delta \) for some choices of \( \tau, \delta \) and sample size in Table 4. The table shows that only for large sample sizes, \( \delta \) around 2–3 and preferably moderate to large \( \tau \) do the data fall reasonably often in case I. Only in case I is the Wald statistic computable, so only for large sample sizes, \( \delta \) around 2–3 and moderate to large \( \tau \) do we find that the Wald statistic is an appropriate tool for statistical inference.

Inference based on the likelihood function is accessible in all cases except those where there is no information in the data about \( \delta \). This occurs only for small sample sizes and with frequencies below 5% (cases V and VI). This is a characteristic of the same–different protocol and not an aspect of likelihood inference.

For each combination of \( \tau, \delta \) and sample size, we simulated 50,000 data sets from the same–different model (1), computed the 95% profile likelihood CI and the 95% Wald CI for \( \delta \) and recorded if the CIs contained the parameter with which we simulated. Table 5 summarizes the resulting coverage probabilities. Note that the standard error of the estimated probabilities in the table is approximately \( \text{se}(\hat{p}) = \sqrt{0.05 \cdot 0.95/50,000} = 0.001 \) providing reasonable accuracy.

Due to the discreteness of the test statistics, coverage probabilities exactly at the nominal level are not possible. However, we can hope for good average performance for interesting values of \( \delta \). Table 5 gives a picture of the coverage probabilities of the Wald and profile likelihood CIs for different choices of sample size, \( \tau \) and \( \delta \). The coverage probabilities for particular combinations of sample size, \( \tau \) and \( \delta \) are specific to those choices and are of less interest – it is rather the trends in the table that are important.

The coverage probabilities are based on those cases that were computable only. The frequencies with which the Wald and likelihood CIs are computable are shown in Table 4.

The coverage probability of the profile likelihood CI is close to the nominal level (95%) for medium-sized \( \delta \) and is in general higher for very low and very large \( \delta \). For the profile likelihood CI there seems to be little dependence on the sample size and the value of \( \tau \). The coverage probability does however approach 90% for \( \delta \) around 3–4, small sample size and large \( \tau \).

The coverage probability of the Wald CI where this is frequently computable (i.e. for \( \delta \) around 2–3 and large sample size) is close to the nominal level. This occurs because the Wald CI closely approximates the profile likelihood CI in these situations. The relative difference between the symmetric approximation and the profile likelihood decreases with the distance from the MLE, so we expect the adequacy of the Wald CI to decrease as \( \tau \) decreases. For instance, we will expect 99% Wald CIs to be farther from the corresponding profile likelihood CIs than 95% Wald CIs are from the corresponding profile likelihood CIs.

A peculiar situation occurs for small sample sizes and \( \delta \) around 2–3. Here the coverage probability is practically 100%, so if the Wald CI is computable, then it will almost certainly contain the true parameter. However, notice that the Wald CI is only computable around half the time in these settings.

The coverage probability of the Wald statistic is generally good for \( \delta \) up to around 4. The coverage probability for the Wald CI for values of \( \delta \) above 4 decreases rapidly for low \( \tau \). This occurs because the MLE of \( \hat{\delta} \) will be infinity with high frequency where the Wald is not computable, so when Wald is computable, the MLE of \( \hat{\delta} \) will be significantly lower than the true \( \delta \) and the Wald CI will only rarely reach high enough to cover the true \( \delta \).

Differences between products might be obvious for large \( \delta \) (say \( \delta > 3 \)), so \( p \)-values and CIs seem less important. However, a key number is the lower confidence limit providing information on the lower limit for likely values of \( \delta \).

Values of \( \delta \) below 2 occur when the product differences are small relative to the perceptual variation. We expect product differences to be small when we are interested in the similarity of two products, so to obtain appropriate \( p \)-values and CIs when similarity is of interest, the Wald statistic is an inappropriate tool.

Small values of \( \delta \) also occur when interest is in product differences. An expert panel may have established a \( \delta \) of intermediate value but subsequent consumer evaluation results in a small MLE of \( \delta \) due to increased perceptual noise for the consumers.

We conclude that Wald-based \( p \)-values and CIs are only appropriate in large samples and for \( \delta \) around 2–3. Likelihood-based inference is possible in small as well as large samples and \( p \)-values can be obtained as long as there is information about \( \delta \) in the data. Likelihood-based inference relieves the analyst from worrying about or examining the adequacy of the Wald statistic.

### Table 5

<table>
<thead>
<tr>
<th>Type</th>
<th>( \tau )</th>
<th>( \delta )</th>
<th>( 0 )</th>
<th>( 0.25 )</th>
<th>( 0.5 )</th>
<th>( 1 )</th>
<th>( 2 )</th>
<th>( 3 )</th>
<th>( 4 )</th>
<th>( 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood</td>
<td>0.5</td>
<td>97.4</td>
<td>97.3</td>
<td>97.4</td>
<td>96.7</td>
<td>94.6</td>
<td>93.6</td>
<td>98.1</td>
<td>98.2</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>97.2</td>
<td>77.9</td>
<td>85.7</td>
<td>93.4</td>
<td>97.0</td>
<td>98.0</td>
<td>94.7</td>
<td>4.41</td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>1</td>
<td>97.5</td>
<td>97.5</td>
<td>97.4</td>
<td>95.5</td>
<td>94.6</td>
<td>94.6</td>
<td>94.6</td>
<td>98.1</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>97.5</td>
<td>79.4</td>
<td>88.9</td>
<td>95.7</td>
<td>95.5</td>
<td>96.9</td>
<td>89.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>2</td>
<td>97.1</td>
<td>97.3</td>
<td>97.4</td>
<td>95.1</td>
<td>94.8</td>
<td>94.8</td>
<td>94.4</td>
<td>93.3</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>97.2</td>
<td>84.1</td>
<td>92.0</td>
<td>95.3</td>
<td>95.1</td>
<td>96.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>( \tau )</th>
<th>( \delta )</th>
<th>( 0 )</th>
<th>( 0.25 )</th>
<th>( 0.5 )</th>
<th>( 1 )</th>
<th>( 2 )</th>
<th>( 3 )</th>
<th>( 4 )</th>
<th>( 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood</td>
<td>0.5</td>
<td>95.9</td>
<td>97.4</td>
<td>97.4</td>
<td>97.4</td>
<td>94.2</td>
<td>93.8</td>
<td>97.8</td>
<td>96.8</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>97.3</td>
<td>75.0</td>
<td>83.3</td>
<td>92.4</td>
<td>97.1</td>
<td>98.8</td>
<td>88.9</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>1</td>
<td>95.9</td>
<td>97.7</td>
<td>97.7</td>
<td>96.3</td>
<td>95.2</td>
<td>94.5</td>
<td>98.1</td>
<td>99.2</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>97.4</td>
<td>77.9</td>
<td>86.8</td>
<td>94.1</td>
<td>97.3</td>
<td>97.6</td>
<td>96.0</td>
<td>62.3</td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>2</td>
<td>97.2</td>
<td>97.3</td>
<td>97.4</td>
<td>95.8</td>
<td>95.1</td>
<td>94.3</td>
<td>94.2</td>
<td>96.8</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>97.5</td>
<td>83.4</td>
<td>91.1</td>
<td>96.3</td>
<td>96.2</td>
<td>94.7</td>
<td>96.6</td>
<td>96.2</td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>20</td>
<td>97.5</td>
<td>95.7</td>
<td>96.6</td>
<td>94.4</td>
<td>95.1</td>
<td>98.1</td>
<td>97.2</td>
<td>99.2</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>97.1</td>
<td>77.2</td>
<td>84.5</td>
<td>97.7</td>
<td>100.0</td>
<td>100.0</td>
<td>98.4</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>1</td>
<td>97.2</td>
<td>97.1</td>
<td>97.3</td>
<td>92.3</td>
<td>97.2</td>
<td>98.4</td>
<td>98.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>95.9</td>
<td>77.9</td>
<td>84.6</td>
<td>91.9</td>
<td>99.6</td>
<td>100.0</td>
<td>94.4</td>
<td>19.9</td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>2</td>
<td>97.3</td>
<td>95.4</td>
<td>95.0</td>
<td>95.5</td>
<td>94.2</td>
<td>90.7</td>
<td>90.4</td>
<td>98.5</td>
<td></td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>95.9</td>
<td>97.8</td>
<td>96.7</td>
<td>99.4</td>
<td>99.5</td>
<td>99.9</td>
<td>96.3</td>
<td>88.7</td>
<td></td>
</tr>
</tbody>
</table>

* Note. same and different samples each equal to \( N/2 \).
5. Inferential procedures for the same–different test (examples)

In this section we return to the examples from the introduction and introduce an example of combining information from several experiments with different protocols.

5.1. Example: Anna’s experiment

Anna was interested in the difference between two products and obtained the data in Table 1. In Fig. 1 we have shown the profile likelihood curve for $\delta$ (solid) for Anna’s data. We have augmented the plot with horizontal lines for the 95% and 99% CIs to ease interpretation.

The curve has its maximum at the MLE ($\delta = 1.88$). The 95% CI indicates that the true $\delta$ is likely to be between zero and 3.21. The profile likelihood curve further shows that values of $\delta$ above 4 have virtually zero support. Also it seems that values of $\delta$ between roughly 1 and 3 are more supported than values between 0 and 1 and that $\delta = 0$ is much less supported than $\delta = \Delta$. Probably the true $\delta$ is nearer 2 than 0.

The $p$-value for the hypothesis of no difference using the likelihood ratio, or equivalently the likelihood root statistic, is $p = 0.0563$, as reported in the introduction. Pearson’s $\chi^2$ test gives $p = 0.0578$ which evidently is very close to that of the likelihood statistics.

It is illustrative to see that if we apply the inaccurate Fisher’s exact test or Yates’ correction in Pearson’s $X^2$ statistic, we obtain $p = 0.119$ from both statistics. This $p$-value is considerably larger than those from the uncorrected Pearson and likelihood statistics.

Suppose Anna believes that a sensory difference between the products, $\delta$ lower than around 1/2 is of no practical interest. The relevant hypotheses are therefore $H_0: \delta < 1/2$ versus $H_1: \delta > 1/2$. The corresponding likelihood-based $p$-value is $p = 0.0723$ which Anna does not consider conclusive evidence of a relevant difference between the products. The Wald statistic misleadingly yields $p = 0.0245$ for the same set of hypotheses. The discrepancy occurs because the symmetric approximation to the profile likelihood curve is inappropriate. This is seen in Fig. 1 where the dashed line represents the symmetric Gaussian approximation assumed by the Wald statistic. The Wald CIs are given by intersections with the horizontal lines, analogous to the likelihood CIs.

5.2. Example: Mark’s experiment

The profile likelihood curve for Marks data (c.f. Table 2) is displayed in Fig. 2. The curve has its maximum at zero since this is the MLE of $\delta$. The profile likelihood curve indicates that $\delta$ is likely to be close to zero. Further, the support for values of $\delta$ decreases as $\delta$ increases and values of $\delta$ larger than 2 are virtually unsupported by the data.

5.3. Example: Combining information from several experiments

Anna previously conducted an experiment with the same two products using the triangle protocol and now she wants to evaluate the information about $\delta$ by using the data from the same–different experiment as well as the data from the triangle experiment. Anna obtained 9 correct answers in 17 triangle tests. The log-likelihood for the triangle protocol is

$$l_{\text{tri}}(\delta; y_{\text{tri}}) = \log f_{\text{tri}}(p, n_{\text{tri}})$$

where $n_{\text{tri}}$ is the number of tests, and $p = p(\delta)$ is the probability of obtaining a correct answer. $p(\delta)$ is given by the psychometric function for the triangle protocol (Brockhoff & Christensen, 2009; Ennis, 1993; Frijters, 1979)

$$p(\delta) = 2 \int_0^\infty \{ \Phi(-2\sqrt{3} + \delta \sqrt{2}/3) + \Phi(-2\sqrt{3} - \delta \sqrt{2}/3) \} \phi(z)dz$$

where $\phi$ is the standard normal PDF. The profile log-likelihood for $\delta$ in the combined same–different and triangle experiment is therefore

$$l(\delta; y_{\text{ad}}, y_{\text{tri}}) = l_{\text{ad}}(\delta; y_{\text{ad}}) + l_{\text{tri}}(\delta; y_{\text{tri}})$$

Notice that the likelihood for the triangle protocol does not contain any nuisance parameters, so to obtain the profile likelihood of $\delta$, we need only profile over $\tau$.

---

* It is a coincidence that this $p$-value is close to the double of the $p$-value from Pearson’s $X^2$ statistic without Yates’ continuity correction. All $p$-values are one-sided.
In Fig. 3 the profile likelihood curve for the triangle data (dotted line), same–different data (dashed line) and the combined data (solid line) is shown.

Observe that the MLE \( \hat{\delta} = 1.73 \) for the combined experiment at the peak of the curve for the combined experiment is in between the MLEs of the individual experiments, as we would expect. The profile likelihood curves for the individual experiments are similar in shape and approximately of equal width reflecting that they carry approximately the same amount of information about \( \delta \). The profile likelihood curve for the combined experiment is narrower reflecting that it contains more information. The profile likelihood curves of the individual experiments separately indicate that the true \( \delta \) is around 2 but also leave some support for values of \( \delta \) close to 0. Notably, the 95% CIs from the individual experiments both contain 0, as seen from Fig. 3. The profile likelihood curve for the combined data shows more evidence of a \( \delta \) around 2 and in particular values of \( \delta \) close to 0 now have considerably less support. This is also reflected in the 95% CI for \( \delta \) \([0.58, 2.62]\) which does not contain 0. However, the likelihood at 0 continues to be too high for the 99% CI to exclude 0 as is evident from the figure.

Notice also that the profile likelihood curve for the combined data is closer to being symmetric than the curves for the individual experiments reflecting that symmetry increases with the sample size. The symmetric approximation assumed by the Wald statistic is therefore better for the combined data than for the individual experiments.

From Fig. 3 we do not expect that \( \delta \) differs between the two experiments, but to illustrate the likelihood ratio statistic suppose the hypothesis of their equality was of interest. The likelihood ratio statistic (4) is LR = 0.0802 with corresponding \( p \)-value of \( p = 0.777 \), so there is virtually no evidence in the data in favour of a difference in \( \delta \) between the two experiments. If the difference, \( \delta_{\text{diff}} = \delta_{\text{tri}} - \delta_{\text{sd}} \) was of particular interest, the likelihood of the joint experiment could have been parameterized in terms of, say, \( \delta_{\text{A}} \) and \( \delta_{\text{B}} \) rather than \( \delta_{\text{tri}} \) and \( \delta_{\text{sd}} \). A profile likelihood curve and likelihood-based CIs could be obtained for \( \delta_{\text{diff}} \) by profiling the likelihood over \( \delta_{\text{B}} \).

### 6. Discussion

We have shown that likelihood inference provides a unified framework for estimation of parameters, CIs and \( p \)-values. Further, we have introduced the profile likelihood curve as an aid in interpreting the information in data about \( \delta \).

The statistical theory of likelihood is well developed but has only to a limited degree entered common practice in many scientific areas including sensometrics and sensory science. We believe this is mainly due to two circumstances. First, the majority of the literature on the theory is accessible mainly to mathematical statisticians and second, the availability of software supporting inference based on likelihood is limited.

In this paper we have tried to make likelihood methods accessible and shown the benefits of such methods for sensory science. Regarding the second matter, we provide the R package sensR.

The inadequacy of the Wald statistic is a general problem whenever a symmetric Gaussian curve is an inadequate approximation to the profile likelihood. Pawitan (2000), for instance, gives an example of the inadequacy of the Wald-based CI and \( p \)-value for a variance parameter in a normal random effects model. The Wald statistic can be viewed as a second-order approximation to the likelihood root statistic, in that it corresponds to the likelihood root statistic when the profile log-likelihood is approximated by a second-order Taylor expansion. Equivalently, the Wald approximation characterizes the profile likelihood by the first two moments: the mean and the variance. When the profile likelihood is inadequately described by the first two moments, we can develop statistics incorporating higher order moments such as skewness and kurto-

The use of likelihood intervals can be seen as a generally applicable way of doing improved statistical inference. This is particularly important for small-sample sizes since most methods usually agree when the sample size is large enough. For a single binomial parameter, it is well known that so-called exact \( p \)-values and exact confidence intervals can be constructed (Clopper & Pearson, 1934; MacRae, 1995), so these can be used for protocols based on a single binomial experiment. However, it is less obvious how this concept extends to protocols based on two binomial experiments such as the same–different and the A–Not A protocols. Boyles (2008) showed that the coverage probability of likelihood intervals is closer to the nominal level on average than the exact and Wald intervals. Further, likelihood intervals also tend to be the shortest.

Brockhoff and Christensen (2009) showed how (profile) likelihood CIs can be obtained for discrimination protocols that can be identified as generalized linear models (GLMs), including the duo-trio, triangle, 2 and 3-APC and A–Not A protocols. The same–different protocol is more complex and is not a GLM, so the convenient methods for GLMs cannot be used here.

Miller (1996) investigated the behavior of the variance of \( \delta \) for the A–Not A protocol, and his Table 2 shows that for \( \delta \) that are not intermediate and obtained from a large sample size the variance of \( \delta \) is an inappropriate measure of the uncertainty in \( \delta \). This is a general problem for inference about \( \delta \), since \( \delta \) is bounded at 0 and for large \( \delta \) there is little information about the location of \( \delta \). Miller also examined several approximate methods to arrive at CIs for \( \delta \) in an A–Not A test. He concludes with the recommendation to compute the CI by several methods to ensure that the CI is not too dependent on the computational method. He found no method to be universally satisfactory. The profile likelihood CI deals with this problem in all test protocols and always provides the CI most supported by the data.

The methods described in this paper are derived assuming that observations are independent. There is a need for models that allow modelling of the distributions of subjects with respect to their discriminations or preferences to cope with heterogeneous populations. Estimation of heterogeneity is possible with replicated data, i.e. when subjects have answered to more than one test. We hope that future research will result in such models.

Another issue that needs further addressing is that of power for both difference and equivalence testing. For instance, how does the power depend on \( \tau \) and the ratio of same and different samples? Analytical power and sample size calculations require knowledge of the distribution of the test statistic under the alternative hypothesis. From the simulations in this paper it is clear that such calculations agree when the sample size is large enough and the likelihood statistic. An appealing alternative is empirical calculations of power based on simulation of the \( p \)-value under the alternative. We provide a function same-diffPwr in our R package sensR that serves this purpose. The price for the simplicity of this method is computational time.

Lee, van Hout, Hautus, and O’Mahony (2007) examined whether assessors can use \( f \beta \) as well as a \( \tau \) criterion in the same–different test. It would be natural to do a comparison of two models based on the two criteria respectively by comparing their likelihoods. We believe the present formulation of the likelihood is a step toward a more general comparison of models with the purpose of assessing the cognitive strategy of the assessors.
7. Conclusions

In this paper, we have discussed likelihood-based inference for a Thurstonian parameterization of the same–different protocol.

We have shown that previously proposed estimators of \( \tau \) and \( \delta \) are in fact ML estimators, and we introduce the profile likelihood curve as a general instrument to obtain inference. We introduce the likelihood root statistic based on the profile likelihood of \( \delta \) as an alternative to the well known Wald statistic to produce \( p \)-values and CIs, and we have shown that the likelihood root statistic is equivalent to the well known \( G^2 \) likelihood ratio statistic for tests of no sensory difference.

We have argued that for hypotheses of no sensory difference the \( G^2 \) likelihood ratio statistic or the asymptotically equivalent Pearson’s \( X^2 \) statistic should be used and that Pearson’s \( X^2 \) statistic with Yates’ continuity correction as well as Fisher’s exact test are available because the variance of the estimated \( \delta \) is often far from 95% whereas the coverage probability of the profile likelihood-based CI is close to the desired 95%. We recommend that inference is based on the likelihood function rather than on the Wald statistic.

We have shown how the likelihood framework can be used to combine information from independent experiments, possibly using different discrimination protocols, to obtain inference for a common \( \delta \).

We provide a free R package with an implementation of the likelihood methodology presented in this paper.

Acknowledgements

The authors are grateful for helpful comments from two anonymous reviewers including one reviewer’s suggestion to show how likelihood methods can be used to combine information from different experiments and the other reviewer’s suggestion for pointing out that the probabilities in Table 4 could be computed as exact binomial probabilities rather than by simulation as we had first done.

Appendix A. R-functions

This section gives a short overview of the functions related to the same–different protocol implemented in the R package sensR. The package is constantly expanding and additional functionality for the same–different protocol may be added with time. The authors are happy to receive contributions and suggestions for improvements. Further information on the package and comprehensive documentation of all functions is included in the package. The package can be downloaded from www.cran.r-project.org/package=sensR.

The main function is \texttt{samediff} which fits a same–different model to data and estimates the MLEs of the parameters. The function returns an object of class \texttt{samediff} for which a number of methods exist.

A.1. Methods for \texttt{samediff} objects and related functions

\texttt{summary} will give a summary of the \texttt{samediff} fit including a table with parameter estimates, confidence intervals and \( p \)-values. Profile likelihood CIs are default, but Wald type CIs can be chosen. \texttt{profile} will compute the profile likelihood of the parameters. There exists a plot method for profile objects that will make reasonable default plots of the profile likelihoods of the parameters as shown in this paper. \texttt{confint} will compute the confidence intervals based on profile likelihoods. \texttt{contour} will make a contour plot of the likelihood surface. By default the normalized likelihood is plotted with confidence limits, but this can be altered by the user. \texttt{plot} will make a plot of the distributions of sensory intensity provided the parameter estimates exist and are positive and finite. \texttt{samediffSim} will simulate data from a same–different model. The function requires the user to specify the parameters, the number of same and different samples and the number of set of observations to simulate.

\texttt{samediffPower} computes the power for a same–different discrimination experiment using profile likelihood inference for a no-difference null hypothesis via simulation. The user must specify the number of samples to simulate, \( \tau \) and \( \delta \) under the alternative, the number of same and different samples and the type I error rate. A number of additional methods exist including \texttt{vcov} that will give the variance–covariance matrix if it exists, \texttt{coef} that will give the parameter estimates, \texttt{logLik} will give the value of the log-likelihood at the MLE, \texttt{AIC} will give the value of AIC for the model fit and \texttt{update} will update the model fit to changes in the data.

Appendix B. Derivation of the MLEs of \( \tau \) and \( \delta \)

The MLEs of \( \tau, \delta \) maximize the log-likelihood function (1) and satisfy the score equations

\[
S(\tau) = \frac{\partial l(\tau, \delta; y)}{\partial \tau} \bigg|_{\tau=\hat{\tau}} = 0 \quad \text{and} \quad S(\delta) = \frac{\partial l(\tau, \delta; y)}{\partial \delta} \bigg|_{\delta=\hat{\delta}} = 0
\]

where the derivatives are given by

\[
\frac{\partial l(\tau, \delta; y)}{\partial \tau} = n_{nsd} \left[ 2 \phi \left( \frac{\tau}{\sqrt{2}} \right) - 1 \right] - \frac{1}{2} \sqrt{2} \phi \left( \frac{\tau}{\sqrt{2}} \right) \phi \left( \frac{\tau}{\sqrt{2}} \right) + n_{nsd} \left[ 2 - 2 \phi \left( \frac{\tau}{\sqrt{2}} \right) \right] \phi \left( \frac{\tau}{\sqrt{2}} \right) + \frac{1}{2} \sqrt{2} \phi \left( \frac{\tau}{\sqrt{2}} \right) \phi \left( \frac{\tau}{\sqrt{2}} \right)
\]

\[
\frac{\partial l(\tau, \delta; y)}{\partial \delta} = n_{nsd} \left[ \frac{1}{\sqrt{2}} \phi \left( \frac{\tau - \delta}{\sqrt{2}} \right) - \frac{1}{\sqrt{2}} \phi \left( \frac{-\tau - \delta}{\sqrt{2}} \right) \right] - \frac{1}{2} \sqrt{2} \phi \left( \frac{\tau - \delta}{\sqrt{2}} \right) \phi \left( \frac{-\tau - \delta}{\sqrt{2}} \right)
\]

Starting with the latter score equation, we find that at the MLEs, assuming they are positive and finite, we have the relation (2). Using this, we find that the last two terms of the first score equation cancel out at the MLE and that the MLE of \( \tau \) is given by (2).

Appendix C. Variance of the \( \delta \) estimator

Following the notation of Bi (2002) the variance of \( \delta \) is given by

\[
\text{Var}(\delta) = \frac{n_{nsd}}{n_{nsd} + 1} \left[ \phi \left( \frac{\tau - \delta}{\sqrt{2}} \right) \phi \left( \frac{-\tau - \delta}{\sqrt{2}} \right) - \phi \left( \frac{\tau}{\sqrt{2}} \right) \phi \left( \frac{-\tau}{\sqrt{2}} \right) \right]
\]
\[ \text{var}(\hat{\delta}) = w^2 \left( \frac{\hat{\mu}_2 \hat{n}_{nd} \hat{\mu}_d}{\hat{n}_{nd}^2 \hat{n}_d} + \frac{\hat{\mu}_2 \hat{n}_{ns} \hat{\mu}_s}{\hat{n}_{ns}^2 \hat{n}_s} \right) \]

where

\[ w = \frac{1}{\sqrt{2}} \left( \frac{\tilde{\mu} - \tilde{\delta}}{\sqrt{\tilde{\mu}}} + \frac{\tilde{\mu} - \hat{\delta}}{\sqrt{\hat{\mu}}} \right) \]

\[ t^2 = \frac{1}{2} \left( \frac{\tilde{\mu} - \tilde{\delta}}{\sqrt{\tilde{\mu}}} + \frac{\tilde{\mu} - \hat{\delta}}{\sqrt{\hat{\mu}}} \right) \]

\[ u = \sqrt{2} \tilde{\mu} \frac{\hat{\delta}}{\sqrt{\hat{\mu}}} \]

and \( \text{var}(\hat{\delta}) \) is given by the diagonal entry corresponding to \( \delta \) of the inverse of the Fisher information matrix evaluated at the MLE, \( \hat{\delta} = (\hat{\mu}, \hat{\delta}) \).

\[ I(\theta) = -\frac{\partial^2 \log(\pi(x, \theta))}{\partial \theta^2} \]

\[ \text{Note that } \text{var}(\hat{\delta}) = \infty \text{ when } \hat{\delta} = 0 \text{ and } \hat{\delta} = \infty \text{ because } w = 0. \]

### Appendix D. Distribution of the ML estimator of \( \delta \)

The distribution of the ML estimator of \( \delta \) in Table 4 is computed as follows:

\[ P(\text{case I}) = \sum_{i=2}^{N-1} \binom{N}{i} P(\hat{n}_d = i) P(\hat{n}_a < i - 1) P(\hat{n}_a > 0) \]

\[ P(\text{case II}) = \sum_{i=1}^{N-1} \binom{N}{i} P(\hat{n}_d = i) P(\hat{n}_a < i) P(\hat{n}_a > 0) + P(\hat{n}_a = 0) P(\hat{n}_a > 0) + P(\hat{n}_a = 0) P(\hat{n}_a > 0) \]

\[ P(\text{case III}) = P(\hat{n}_a = 0) P(\hat{n}_a > 0) P(\hat{n}_a > 0) \]

\[ P(\text{case IV}) = P(\hat{n}_a = 0) P(\hat{n}_a > 0) P(\hat{n}_a > 0) + P(\hat{n}_a = 0) P(\hat{n}_a > 0) \]

\[ P(\text{case V}) = P(\hat{n}_a = 0) P(\hat{n}_a = 0) \]

\[ P(\text{case VI}) = P(\hat{n}_a = 0) P(\hat{n}_a = 0) \]

The individual terms are given by the binomial PDF and CDF functions.

### References


Appendix B

Thurstonian models for sensory discrimination tests as generalized linear models

1. Introduction

Sensory discrimination tests such as the triangle, duo-trio, 2-AFC and 3-AFC tests together with the A-not A and same–different tests produce binary data leading to count data when aggregated over replications and/or assessors/consumers. Hence, the basic statistical methods needed for planning such experiments and analysing such data as they come, can be found among methods based on the binomial distribution and standard methods for analysing tables of counts. The former include basic binomial-based power and sample size calculations and exact binomial hypothesis testing, see e.g. Schlich (1993). The latter could be using the Pearson’s chi-square or McNemar’s test for paired 2-by-2 tables, see e.g. Bi (2006). The weakness of this approach working on the count scale is that it is test protocol dependent: the number of expected correct answers for the same products depend heavily on which test that is carried out. This has been pointed out by several authors, see e.g. Ennis (1993a), and amounts to the lack of a common framework for comparing the underlying sensory differences across different testing paradigms. The Thurstonian approach of transforming the number of correct answers into an estimate, called d-prime (d'), of the underlying (relative) sensory difference, is the solution of this deficiency of the count data statistical approach.

Generalized linear models, McCullagh and Nelder (1989), are extensions of linear models (regression, analysis of variance and combinations thereof) designed to cope with non-normal data including binary and count (e.g. Poisson) data. The observed data are linked to a linear model structure by a non-linear function. The benefit of the approach is that the whole statistical apparatus of linear modelling carries over to the more general non-normal/non-linear setting. So apart from a theoretical framework, it provides a practical approach, through existing software, to analyse binary/count data of various kinds with the same level of detail (ANOVA, Regression, ANCOVA etc.) as more commonly known to be carried out for quantitative data with normally distributed errors.

It becomes quite obvious below that the Thurstonian approach to handling sensory difference testing will, at least in some occasions, amount to a generalized linear model approach but with the so-called links defined by the Thurstonian psychometric functions rather than the traditionally used links. There exists several statistical packages with generalized linear model features that allows the user to specify special link functions as needed. It is therefore possible to explore this intersection between statistical theory and practice and the Thurstonian approach of sensory discrimination testing. This has not been done before, and we show that it does contribute non-trivially to the ability to analyze sensory difference data.

In the next section the generalized linear models are introduced and motivated in the context of sensory data. Then two protocol specific sections follow: One on the four classical protocols: 2-AFC, 3-AFC, duo-trio and triangle and one on the A-not A protocol. Each section contains theoretical expressions of the models for the protocols, the relation to the generalized linear models is clarified and a simple example is given. The R-package is described in the appendix together with a short description of how to get started with the R-programme.
2. General and generalized linear models

In this section the generalized linear models are introduced and motivated by first introducing the general linear model. The general linear model is a theoretical framework that in a unified way covers many basic kinds of statistical analyses; (fixed effects) analysis of variance (ANOVA), (multiple) linear regression (MLR) and analysis of covariance (ANCOVA), (see e.g. Crawley, 2005; McCulloch & Searle, 2001). Any such classical analysis can be seen as a result of fitting a linear model to the data $y_i, i = 1, \ldots, n$ given by

$$y_i = x_i' \beta + e_i$$

where $x_i$ is a vector of design variables for the $i$th observation and $\beta$ is the corresponding vector of unknown parameters. The components of $x_i$ are either quantitative values (as in MLR) or zero-one dummy codings of qualitative information (as factors in ANOVA). Components of $\beta$ are either real regressions coefficients or an effect of a qualitative treatment level. Both type of components may be present (as in ANCOVA). The “linear” name comes from the fact that the expected structure in the data is expressed as a linear function of the parameters in the model:

$$E(y_i) = x_i' \beta$$

Note that this does not exclude the possibility of modelling non-linear relations between quantitative $x$ and $y$ by the use of, e.g. polynomial regression models.

It is common to supplement with an assumption of observations being independent and homoscedastic normally distributed:

$$e_i \sim N(0, \sigma^2)$$

and the theory is build around this assumption, although most practical statistical procedures will be valid at least approximately in much greater generality due to the Central Limit Theorem.

In short, the outcome of an application of a version of the general linear model is always essentially two things:

1. The estimated parameter values $\hat{\beta}$.
2. The uncertainty information about these estimates Cov($\hat{\beta}$).

There are direct and simple matrix-based formulas for computing these two things, which may be found in textbooks covering the general linear model, see e.g. McCulloch and Searle (2001).

From this, $t$-tests for the significance of individual or simple combinations of parameters can be found. Also $F$-tests for composite hypotheses often employed in ANOVA can be deduced from this. The estimates of the parameters $\beta$ are least squares estimates and under the assumption of normal homoscedastic errors, they are also maximum likelihood estimates (MLE).

However, for clearly non-normal data, for instance categorical/ binary data, the methods based on the general linear model cannot in general be trusted. Fortunately, there is an even more general family of methods, the generalized linear models, that includes the normal linear model as a special case and also allows for non-normality, categorical responses, non-homogeneous variances and some kind of non-linearity in the data structure. Let us assume that we observe a binary response $y_i$, for which we can assume that the basic observations come from a binomial distribution with $n = 1$ and $p_i$, as the probability parameter:

$$y_i \sim \text{bin}(1, p_i)$$

This could be the responses from a number of triangle tests $i = 1, \ldots, n$. Also assume that different stimulus concentrations $x_i$ were applied and that the consumer panel background information, e.g. gender, is of particular interest. Had the response been quantitative, e.g. sweetness intensity, we could have used a general linear model with gender as a qualitative descriptor and the stimulus concentration as a quantitative descriptor, and express a model where the response could depend (linearly) on the stimulus concentration, possibly depending on the gender, that is, a model with the two “independent” variables gender (qualitative) and stimulus concentration (quantitative) and their interaction:

$$E(y_i) = \beta_0 + \beta_1 x_i + \beta_2 (\text{gender}) + \beta_3 (\text{gender})x_i$$

(1)

As it seems definitely questionable to set up such a quantitative model directly for the binary zero-one outcomes $y_i$ , it could be somewhat less artificial, perhaps, to do so for the expected value of the outcome, as actually expressed in (1). Recall, that the expected value for a binomial $\text{bin}(1, p_i)$ outcome is given by $p_i$, that can take any value between zero and one, so theoretically $p_i$ is a quantitative and continuous construct and a model relating $p_i$ to the stimulus concentration and gender factor is what is sought here. However, there is still a theoretical flaw in (1): The right hand side quantitative expression can in principle attain any value including negative as well as values larger than 1, whereas the left hand side expected value only can attain values between 0 and 1. This theoretical conflict can be avoided by linking the linear model to a suitable non-linear function of the expected values instead, for instance the logit-function:

$$\log\left(\frac{E(y_i)}{1 - E(y_i)}\right) = \log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_i + \beta_2 (\text{gender})$$

(2)

The logit-function maps values between 0 and 1 into the full set of real valued numbers. This model is known as a logistic regression and/or log-odds model and has a history of its own, Cramer (2002). The generalized linear model (GLIM), see McCullagh and Nelder (1989) is a theoretical and methodological framework that includes the logistic regression model, the classical linear normal model, and many other models depending on the specific choice of the two concepts:

- The distribution of $y_i$.
- The link between $E(y_i)$ and the linear model structure.

The logistic regression model is a GLIM with the binomial distribution and the logistic link. A probit regression model, cf. Cramer (2002), is a GLIM with the binomial distribution and the probit (inverse standard normal) link. A classical linear normal model is a GLIM with the normal distribution and the identity link. The most well-known GLIMs, that is, those that are commonly available in software, includes the normal, binomial, poisson and gamma distributions and the identity, logit, probit, log, complementary loglog, square-root and inverse links.

The benefit of such a framework is that it offers an approach for the analysis of data that can be used for all these situations together with the theoretical foundation for the analysis. The results of a GLIM is essentially as for the linear model:

1. The estimated parameter values $\hat{\beta}$.
2. The uncertainty information about these estimates Cov($\hat{\beta}$).

There are generally no closed form solutions for these two pieces of information. However, maximum likelihood estimates can be obtained by a simple algorithm, the Iterative Weighted Least Squares (IWLS) algorithm. Essentially, the algorithm repeatedly applies a weighted linear model to a transformed version of the data. However, this is an integrated part of GLIM software and the end user does not have to know the details of this. The main message here is that we can handle the analysis of non-normal data following exactly the same principles as we would use for a classical normal based analysis, at least whenever the relevant model can be expressed as a GLIM.
3. The triangle, duo-trio, 2-AFC and 3-AFC as generalized linear models

A common feature for the four sensory difference tests mentioned above is that the observation consists of a single number of correct answers, $y$, that can be assumed to follow a binomial distribution $\text{bin}(n,p)$. For the no difference hypothesis test nothing more is needed, the classical binomial based statistical methods can be used. In Thurstonian modelling the focus is on quantifying/estimating the underlying sensory difference $\delta$ between the two products that are compared in the difference test. This is done by setting up mathematical models for the cognitive decision processes that are used by assessors in each sensory test protocol, see e.g. Ennis (1993a). In this way the observed number of correct answers is directly linked to the underlying sensory difference $\delta$ through a protocol specific non-linear function, the so-called psychometric function. The psychometric functions used in this section are found in Ennis (1993a), but all of them have origins going much further back in time, for the so-called 2-AFC back to Thurstone (1927). It is beyond the scope of this paper to give a full review of the history of all of these methods, but a few key references are given here. In Bock and Jones (1968) probably the first systematic exposition of the psychological scaling theory and methods by Thurstone was given. This included a sound psychological basis as well as a statistical one, with a for that time modern inclusion of the use and assessment of maximum likelihood methods. Within the field known as signal detection theory, see e.g. Green and Swets (1966) or Macmillan and Creelman (2005), methods of this kind were further developed, originally with special emphasis on detecting weak visual or auditory signals. Further developments of such methods and their use within food testing and sensory science have developed over the last couple of decades with the numerous contributions of D. Ennis as a corner stone. A few of these are: Ennis (2003, 1993b) and Ennis et al. (1998, 1998).

For the m-AFC method the psychometric function is given by:

$$f_{m-AFC}(\delta) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(z - \delta)^2}{2} \right] \phi(z)^{m-1} dz$$

where $\phi$ is the standard normal distribution function. So for $m = 3$ it becomes:

$$f_{3-AFC}(\delta) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(z - \delta)^2}{2} \right] \phi(z)^2 dz$$

$$= \int_{-\infty}^{\infty} \phi(z - \delta) \phi(z) dz$$

And for $m = 2$ it simplifies to:

$$f_{2-AFC}(\delta) = \phi(\delta / \sqrt{2})$$

For the triangle method the psychometric function is:

$$f_{\text{tri}}(\delta) = 2 \int_{0}^{\infty} \left\{ \phi \left[ -z \sqrt{3} + \delta \sqrt{2} / 3 \right] + \phi \left[ -z \sqrt{3} - \delta \sqrt{2} / 3 \right] \right\} \phi(z) dz$$

And the duo-trio psychometric function is given by:

$$f_{\text{dt}(\delta)} = -\phi(\delta / \sqrt{2}) - \phi(\delta / \sqrt{6}) + 2 \phi(\delta / \sqrt{2}) \phi(\delta / \sqrt{6})$$

Operationally these functions are used in their inverse versions, since given an observed fraction of correct answers $\hat{p} = y/n$ the value of delta, called $d'$ (d-prime), that fits with the psychometric function:

$$f_{\text{ml}}(d') = \hat{p}$$

is determined. The psychometric functions for the four protocols are shown in Fig. 1. Also shown is the function corresponding to the logistic link (dashed). The theoretical model version of this operation together with the basic binomial assumption can be expressed in the following way, where $y_i$ is the answer given in the $i$th test:

$$y_i \sim \text{bin}(1, p_i), \quad g(p_i) = \delta$$

It is now evident that this is a GLIM with the binomial distribution and the inverse psychometric function $f_{\text{ml}}(\delta) = g(\delta)$ as the link—compare with Eq. (2). The linear structure of this model is the simplest possible: a model with only an “intercept” parameter: $\delta$.

Some statistical software for GLIMs have the option of allowing for user defined link-functions, so the practical consequences of identifying the difference test situation as a GLIM is that the computation of $d'$ becomes the estimation of the intercept in a GLIM. Hence, the variance of the $d'$ is given for free since it is given by the standard error of the intercept estimate, which is automatically provided by the GLIM approach. Also, when $d'$s for different product versions are found with the purpose of comparing these, the comparison can be carried out as a standard linear contrast investigation in a GLIM in analogy with a classical two-sample $t$-test. Moreover, frameworks where more than one or two experimental settings are carried out, calling for several different $d$-prime calculations and comparisons or subsequent modelling/investigation of these, can be incorporated into a single joint analysis of the data in a usual $t$-test, ANOVA or regression manner. All this will be exemplified in the following. Note that this is no different from what is usually recommended in these situations, (see e.g. Bi, Ennis, & O’Mahony, 1997). The novelty lies in the fact that the situation is embedded into a classical (generalized) linear model setting which means that roughly all that is covered in Bi et al. (1997) becomes standard (generalized) linear model methodology.

We have included the definition of the link functions and other GLIM attributes in Appendix B for a complete specification.

Another novelty is the fact that an implementation of everything covered in this paper is available for everyone to download from http://www.cran.r-project.org/package-sensR/ in terms of the R-package sensR, see the Appendix. And since R is Open Source software everyone can obtain it for free at http://www.r-project.org.

![Fig. 1. Psychometric functions for the four discrimination tests. The logistic link function is also shown (dashed).](image-url)
3.1. Examples

We start with a simple example. Assume that we have 10 out of 15 correct responses in one of the four discrimination test settings covered here. For the triangle and 3-AFC setting this would amount to a one-tailed exact binomial \( P \)-value for the hypothesis of no difference of 0.0085, cf. e.g. the tables of Schlich (1993). For the duo-trio and 2-AFC settings the \( P \)-value is 0.1588. Using the generalized linear model procedure of the R-software together with the four psychometric families as given in the R-package sensR, we obtain standard linear model statistical output as shown in Table 1.

It is not the intention here to teach the reader the syntax of R nor the specifics of the GLIM procedure of R. It should be clear, however, that the output as presented in Table 1 is of standard linear model type as would be given by any statistical programme: The estimate, its standard error, a statistic for the hypothesis that the parameter is zero (parameter value divided by its standard error) and finally a \( P \)-value for this hypothesis test. For comparison, the results from a standard logistic regression is given also.

The \( P \)-values coming from this standard GLIM analysis should not be used, however. Primarily, because they as default in linear and generalized linear model software are given as two-tailed \( P \)-values. For the discrimination testing situation here we should use one-tailed \( P \)-values and there is no reason not to use the exact binomial based \( P \)-values. For this reason a procedure was constructed such that only the proper and relevant information for the simple discrimination test analysis is given and such that the GLIM syntax of R is hidden for the user. Due to its simplicity the exact R-command is given here: At the R command prompt simply write: `discrim(10,15, "triangle")` and the \( d \), the standard error and the proper \( P \)-value is given together with confidence limits of \( d \).

The package can also be used for power and sample size computations, (Ennis, 1993a; Schlich, 1993). For instance the power of (probability of detecting) \( d = 1 \) with \( n = 30 \) tests using a level 0.05 hypothesis test is found as `discrimPwr(1,30,,05,"triangle")`. Similarly the number of samples needed for obtaining a power of 90\% for \( d = 1 \) using a level 0.05 hypothesis test is found as `discrimSS(1,9,05,"triangle")`. The result for the triangle test as well as for three other tests are given in Table 2.

These functions can easily be used also by users new to R. Experienced R-users may use these as background for various plotting of the basic psychometric functions. A simple discrimination test simulation device is also provided.

We now turn to a more involved example, showing how we may combine a traditional statistical approach with the psychometric foundation of Thurstonian modelling. Assume that four product samples of increasing concentrations each were tested by two groups of subjects, say, males and females, yielding the data seen in Table 3.

So for this study, 160 subjects (consumers) were used. The most natural choice of analysis for a statistically minded person would be a binomial based modelling, typically the logistic regression within the GLIM framework. The structure of the experiment and the samples enables a decomposition of the information into the following effects: main effects of gender (1 degree of freedom (DF)), linear trend effect of Sample concentrations (1 DF), remaining sample differences (2 DF), interaction between linear trends and gender, i.e. different slopes for genders (1 DF) and the remaining sample-gender interaction (2 DF). Formally, the full model could be expressed as

\[
\logit(p_{ij}) = \log\left( \frac{p_{ij}}{1-p_{ij}} \right) = \beta_0 + \beta_1\text{Conc}_i + \gamma_{ij}
\]

where \( p_{ij} \) is the probability of success for the \( n_j \) subjects of gender \( j \) for sample type \( i \). The results of such an analysis can be put into a so-called analysis of deviance (ANOVA) table similar to an ANOVA table. In Table 4 this is found under the logistic heading.

Each deviance value expresses the difference between having the effect in the model or not, sequentially from the top. The difference is measured by minus twice the difference in log-likelihoods and standard statistical theory calls for using a \( \chi^2 \)-distribution for approximate \( P \)-values. In Table 4 the results of doing a triangle and 3AFC based analysis of the data corresponding to the following Thurstonian generalized linear models:

\[
\begin{align*}
\text{GTriangle}(p_{ij}) &= \beta_0 + \beta_1\text{Conc}_i + \gamma_{ij} \\
\text{GThreeAFC}(p_{ij}) &= \beta_0 + \beta_1\text{Conc}_i + \gamma_{ij}
\end{align*}
\]

where \( \text{Gtriangle} \) and \( \text{GthreeAFC} \) are the proper link functions, that is, the inverted psychometric functions. Based on the implementations of these in the R-package sensR, these additional analyses can be handled similar to the standard logit-based analysis, see the appendix for details. It is clear from the results in Table 4 that when it comes to the conclusions regarding the structure of the underlying pattern of effects it makes hardly any difference what link function is used in this case. The conclusion is rather clear: There is a main effect of gender and a linear increase of sensory difference related to the sample concentration. All other possible effects are clearly non-significant. It is also clear that the full structured model in each case is

### Table 1

<table>
<thead>
<tr>
<th>Test</th>
<th>Power</th>
<th>Sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle</td>
<td>.2330</td>
<td>283</td>
</tr>
<tr>
<td>3-AFC</td>
<td>.9542</td>
<td>23</td>
</tr>
<tr>
<td>2-AFC</td>
<td>.2283</td>
<td>30</td>
</tr>
<tr>
<td>Duo-trio</td>
<td>.9173</td>
<td>319</td>
</tr>
</tbody>
</table>

### Table 2

Power and sample size computations for four discrimination tests—see the text for details.

#### Table 3

Data for involved example.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Males</th>
<th>Females</th>
<th>Males</th>
<th>Females</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correct</td>
<td>Total</td>
<td>Correct</td>
<td>Total</td>
</tr>
<tr>
<td>A (Conc = 1)</td>
<td>9</td>
<td>20</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>B (Conc = 2)</td>
<td>11</td>
<td>20</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>C (Conc = 3)</td>
<td>13</td>
<td>20</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>D (Conc = 4)</td>
<td>14</td>
<td>20</td>
<td>18</td>
<td>20</td>
</tr>
</tbody>
</table>

### Table 4

Standard output from 10 correct out of a total of 15 responses.

<table>
<thead>
<tr>
<th>Test</th>
<th>Estimate</th>
<th>SE</th>
<th>z-Value</th>
<th>P-value (2-tailed normal)</th>
<th>P-value (1-tailed exact)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle</td>
<td>2.3214</td>
<td>0.6510</td>
<td>3.566</td>
<td>0.0004</td>
<td>0.00085</td>
</tr>
<tr>
<td>3-AFC</td>
<td>1.1159</td>
<td>0.4359</td>
<td>2.580</td>
<td>0.0105</td>
<td>0.0085</td>
</tr>
<tr>
<td>2-AFC</td>
<td>1.5189</td>
<td>0.7159</td>
<td>2.122</td>
<td>0.0339</td>
<td>0.1509</td>
</tr>
<tr>
<td>SensR</td>
<td>0.6931</td>
<td>0.5477</td>
<td>1.266</td>
<td>0.206</td>
<td></td>
</tr>
</tbody>
</table>
really just the same as having eight different binomial distributions, and hence exactly the same model, no matter what link function is allowed for the two lines are heavily overlapping, expressing the fact that the likelihood is highly asymmetric as we shall show shortly. As noted by Bi et al. (1997), computing Wald based CIs for this formulation, the A-not A method becomes:

\[ d' = z_{\text{not-A}} - z_{\text{A}} \]

Hence, the A-not A test can be handled by arranging the data properly and using the probit link function—one of the standard links in any GLIM software.

4.1. Example

Suppose we conducted a A-Not A test and obtained the data shown in Table 6.1.

If we assume the data are independent, we may estimate \( \delta \) using the above mentioned A-Not A method of the R-package sensR. We obtain the usual estimate \( d' = 0.378 \) along with the standard error \( \text{se}(d') = 0.178 \) and the P-value = 0.0237. This estimate of \( d' \) coincides with that given by Bi and Ennis (2001b). The estimates of the standard errors are close (Bi & Ennis (2001b) obtain \( \text{se}(d') = 0.178 \)) whereas the p-values differ slightly (Bi & Ennis (2001b) obtain P-value = 0.034 based on Pearson’s \( \chi^2 \) statistic), hence their estimate is slightly more conservative.

5. Likelihood and confidence intervals

Another advantage of the identification of the discrimination models as generalized linear models is that likelihood based confidence intervals (CIs) are easy to compute. The main advantages of the likelihood CIs over the classical normal based symmetric Wald CIs are that they need not be symmetric reflecting more properly the evidence about the parameters, and that they are defined in extreme situations such as when all samples are answered correctly. As noted by Bi et al. (1997), computing Wald based CIs for \( \delta \) is not problematic. This happens because the likelihood is highly asymmetric as we shall show shortly.

The theory of likelihood and likelihood CIs will not be explained here. The topics are introduced in the context of same–different tests in Christensen and Brockhoff (in review) and the concepts carry over to the test protocols considered here. For the statistically significant difference, we can use the likelihood CIs to estimate the lower and upper bounds of the \( \delta \) value.
interested reader we recommend the book by Pawitan (2001) as a very good introduction to likelihood inference.

Consider first the situation that we in a triangle test obtain 10 correct and 10 incorrect answers. The likelihood for this test is shown in Fig. 3 as the solid line. The 95% and 99% likelihood CIs are given by the intersection with horizontal lines. These lines follow from the asymptotic $\chi^2$-distribution of twice the relative or normalized log likelihood of one parameter. The level of the lines are given by $\exp\left(-F(1 - x)/2\right)$, where $F()$ is the cdf of a $\chi^2$-density with one degree of freedom. For $x = (0.05, 0.01)$, the level of the lines are at (0.1465, 0.0362).

In package MASS (Venables & Ripley, 2002) for R, a function `confint` to compute likelihood confidence intervals for generalized linear models is implemented. We can make use of this function via the implementation of the link functions for the discrimination test protocols in package sensR to obtain the likelihood CI for $\delta$ in the triangle test; (0, 2.55). This is clearly another advantage of the identification of the discrimination test protocols as generalized linear models.

The Wald confidence interval is based on a normal approximation to the likelihood curve. We have shown this symmetric approximation to the likelihood in Fig. 3 with the dashed line. The Wald CI is in this case (0.317, 2.62) and leads to the wrong conclusion that $\delta$ is significantly different from 0 at the 95% level. The truth is that the likelihood of a true underlying $\delta = 0$ is relatively high and reasonably supported by data.

The Wald CI is obvious only reliable when the likelihood curve is symmetric. We have found that the likelihood is often asymmetric for the duo-trio and triangle tests and reasonably symmetric for 2 and 3-AFC tests and for A-Not A tests. The symmetry is generally better for $\delta$ of intermediate size (i.e. around 2–3) and the degree of symmetry increases with the sample size. It is clear that for equivalence testing (Bi, 2005, 2007; Ennis, 2008; Meyners, 2007; Schuurmann, 1987), where $\delta$ is often close to 0, Wald tests and CIs are unreliable and likelihood modelling will provide appropriate and relevant inference.

In Fig. 4 the likelihood is shown for a duo-trio test with 10 correct and 12 incorrect answers. In such a situation, where the fraction of correct answers is lower than the guessing probability the estimate of $\delta$ is 0 and symmetric Wald based CIs are inappropriate. An appropriate 95% likelihood confidence interval is however given by (0, 1.5) as seen in the figure. In equivalence testing the true $\delta$ is often near 0 and the situation just described is not uncommon.

Another common situation, when the sample size is small to moderate and the true $\delta$ is high occurs when all answers are cor-

---

**Table 6**

<table>
<thead>
<tr>
<th>Sample</th>
<th>Response</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>57</td>
<td>43</td>
</tr>
<tr>
<td>Not-A</td>
<td>42</td>
<td>58</td>
</tr>
</tbody>
</table>

---

**Fig. 2.** The effect of concentration on d-prime for each gender. 95% confidence intervals shown for males (dashed) and females (dotted).

**Fig. 3.** Likelihood (solid) and the normal approximation (dashed) for a triangle test with 10 correct and 10 incorrect answers. Intersection with horizontal grey lines denote the 95% and 99% likelihood confidence intervals (solid) and Wald confidence intervals (dashed).

**Fig. 4.** Likelihood for a duo-trio test with 10 correct and 12 incorrect answers. Intersection with horizontal grey lines denote the 95% and 99% likelihood confidence interval.
rect. The MLE of $\delta$ is now $\infty$, but we can still produce a likelihood and corresponding CI. We have shown a likelihood for the duo-trio test where 10 out of 10 answers were correct in Fig. 5. Note that even with this small sample size, we find that the 95% likelihood CI is $(2.54, \infty)$.

In our package, sensR, we have included confint and profile methods facilitating the use of (profile) likelihood methods and corresponding CIs.

### 6. Summary and discussion

We have clarified the practical, technical and scientifically relevant connection between certain Thurstonian models used in sensory and signal detection science and the statistical concept of generalized linear models. This covers the duo-trio, triangle, 2-AFC, 3-AFC and A-Not A test protocols. We have used this connection to implement the basic $d$-prime calculations for these protocols in the R-package sensR by implementing the corresponding psychometric functions and their inverses. For the basic $d$-prime calculation this provides nothing new in the sense that it gives the proper (maximum likelihood) $d$-prime results well known previously from the literature. The novelty here lies in providing a free and open source software package to do the discrimination test analysis including power, sample size and simulation procedures. And a package embedded in the internationally expanding R-environment. A real benefit is that the $d$-prime uncertainty computation becomes an integral part of the statistical analysis and even more so the fact that it is now possible to directly combine standard statistical analysis such as ANOVA and regression analysis with Thurstonian modelling in a natural and optimal way.

The generalized linear models have previously been brought up as a tool for analysing sensory data, (Brockhoff & Müller, 1997; Hunter, Piggot, & Lee, 2000) but only in their classical versions not specifically taking the Thurstonian approach. A discussion of classical GLIM links versus Thurstonian links was brought up in Brockhoff (1995). In Brockhoff (2003) ordinary and corrected GLIMs were discussed as a tool to handle the replicated situation, but still in their classical versions. The developments of the current paper stand as the basis for formulating formal Thurstonian versions of models for replicated discrimination tests as an alternative to the commonly used beta-binomial approach, cf. (Bi & Ennis, 1998; Ennis & Bi, 1998). This is ongoing work. Also ongoing is the further developments of this approach to handle A-Not A protocols with sureness leading to ordinal data with and without replications.

### Appendix A. Implementation in R

R is an Open Source implementation of the well-known S language. R is a language and environment for statistical computing and graphics freely available for anyone and easily downloadable from the R-project home page [http://www.r-project.org/](http://www.r-project.org/). It is not the intention to introduce the use of R here. Introductory material can be found on the website and Dalgaard (2002) gives a good introduction with focus on statistical issues, while (Venables & Ripley, 2002) is a comprehensive reference. A so-called R-package named sensR was constructed by the authors and this package can be downloaded from [http://cran.r-project.org/package=sensR/](http://cran.r-project.org/package=sensR/) or will be emailed by the authors if requested. The package includes full documentation, help-files and examples in an R-integrated fashion.

The remaining part of this appendix will briefly describe the R-code used in this paper.

#### A.1. R-code for the examples in Section 3.1

We assume that you have R up and running and that you have installed the package sensR on your computer. To load the package and make the tools presented in this paper available, simply at the R-prompt type

```r
> library(sensR)
```

With the package loaded, we obtain the simple discrimination, power and sample size calculations with

```r
> discrim(success = 10, total = 15, method = "triangle")
```

Next we show how to fit the model with gender and concentration as explanatory variables. We only show the final model, but the model is easily extended:

```r
> data <- expand.grid(conc = 1:4, gender = c("Male", "Female"))
> data$correct <- rep(20, 8)
> data$total <- rep(28, 8)
> model <- glm(cbind(correct, total - correct) ~ gender + conc, + data, family = triangle)
> summary(model)$coefficients
```

Fits of models with other links are obtained by exchanging `triangle` with e.g. `threeAFC` for the 3-AFC method and `binomial` for the logit method.

---

**Fig. 5.** Likelihood for a duo-trio test with 5 correct and 0 incorrect answers. Intersection with horizontal grey lines denote the 95% and 99% likelihood confidence interval.
A.2. R-code for examples in Section 4.1

To fit the A-Not A model, we use the `AnotA` function

```r
> AnotA(x1 = 57, n1 = 100, x2 = 5, n2 = 100)
```

Call: `AnotA(x1 = 57, n1 = 100, x2 = 5, n2 = 100)`

Results for the A-Not A test:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>Lower</th>
<th>Upper</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>d-prime</td>
<td>0.3792676</td>
<td>0.1784076</td>
<td>0.02865908</td>
<td>0.7279402</td>
</tr>
</tbody>
</table>

A.3. The psychometric functions

The psychometric functions are implemented as inverse link-functions in so-called family objects. One can (as we did) use these for plotting purposes and we will show how to obtain a simple plot (which is not shown due to space requirements) of the psychometric function for the duo-trio test:

```r
> x <- seq(-0.5, length = 100)
> y <- duotrio(signKey(x))
> plot(x, y, type = "l", ylim = c(0, 1), ylab = "probability")
```

A.4. Profile likelihoods

The profile likelihood for the triangle experiment in Fig. 3 can be made with the following command

```r
> plot(profile(discrim(10, 0.5, method = "triangle")))
```

Further information on how to compute and plot the profile likelihoods in the end situations is found in the help pages (e.g. `?duotrio`) and examples are given, which can be run with for instance example(`duotrio`).

A.5. Additional features

The package sensR currently also contains functions to analyze same–different tests as well as perform ROC (Receiver Operating Characteristic) curve analysis of A-Not A type experiments including AUC (area under the ROC curve) computations.

The package is continuously evolving with more features and will include functions to support the developments mentioned in the summary and discussion. The authors are happy to receive comments and suggestions for changes or additional features and functions.

Appendix B. GLIM attributes for discrimination tests

B.1. Model and likelihood

The model for a discrimination test assuming independent data is given in Eq. (8). If other effects might influence discrimination, they can be included in the model and we may define the more general model

\[ g(\pi) = \eta = X\beta \]

where \( \eta \) is the general linear predictor, \( X \) is a \( n \times p \) design matrix and \( \beta \) is a vector of parameters.

The log likelihood function equals the log density function and for binomial data it is given by

\[ l(\delta; y_i) = \log f(y_i) = w_i m_i \left( \frac{\pi}{m_i} \log \frac{\pi}{1 - \pi} + \log (1 - \pi) \right) + \log \left( \frac{m_i}{m_i y_i} \right) \]

where \( y_i^* = y_i / m_i \) is the scaled response and \( w_i \) are prior weights. The additive constant is typically excluded from the definition of the log likelihood, since the log likelihood is only meaningful up to an additive constant anyway.

The deviance components are defined as

\[ d_i = 2w_i \left( \log y_i^* - \log \left( \frac{\pi}{m_i} \right) \right) \]

and the total deviance is

\[ D = \sum_{i=1}^{n} d_i \]

B.2. GLIM attributes

The parameter estimates \( \hat{\beta} \) can be computed using iterative weighted least squares (IWLS) or a more direct optimization of the likelihood function with respect to the parameters. The IWLS equation reads

\[ X'WX\beta = X'Wy \]

which is solved iteratively for \( \beta \) by updating the weights \( W \) and the working response \( Z \) in each iteration.

The weight matrix is diagonal with elements given by

\[ W_i = w_i \left( \frac{\partial V(\pi_i)}{\partial \pi_i} \right)^{-1} \]

where \( V(\pi_i) = \pi_i(1 - \pi_i) \) is the variance function for binomial data, i.e. \( \text{var}(y_i) = \phi V(\pi_i) \) and the derivative depends on the specific discrimination test.

The working response is given by

\[ Z_i = \eta_i + \frac{\partial V(\pi_i)}{\partial \pi_i} (y_i - \pi_i) \]

The variance of \( \hat{\beta} \) can be approximated with the expected Fisher information matrix evaluated at the MLE, \( X'WX \), hence

\[ \text{var}(\hat{\beta}) = (X'WX)^{-1} \]

The inverse link functions map from the \( \delta \) scale to the probability scale and are given by the psychometric functions in Eqs. (3)–(8)

The derivatives \( \frac{\partial}{\partial \delta} f_{\text{ARC}}(\delta) \) are 0 for non-positive \( \delta \), whereas for positive \( \delta \) they are given by

\[ \frac{\partial}{\partial \delta} f_{\text{ARC}}(\delta) = \frac{2}{3} \sqrt{3} \int_0^\infty \left( \phi \left( \frac{\sqrt{3}}{2} + \delta \sqrt{3} \right) - \phi \left( \frac{\sqrt{3}}{2} - \delta \sqrt{3} \right) \right) \phi(z)dz \]

For a general psychometric function, \( f_{\text{ARC}}(\delta) \), the inverse link function maps from the probability scale to the \( \delta \) scale and is given by

\[ \delta = f_{\text{ARC}}^{-1}(\pi) = \begin{cases} \text{root of } (f(\delta) - \pi) = p_0 & p_0 < \pi < 1 \\ 0 & 0 < \pi < p_0 \end{cases} \]
to which there is seldom an analytical solution. Here \( p_0 \) is the chance probability for \( f_{ps}(t) \), i.e. \( p_0 = 1/2 \) for the duo-trio and 2AFC and \( p_0 = 1/3 \) for the triangle and 3AFC. For the 2AFC function, the inverse can be explicitly expressed as

\[
\delta = \frac{\pi}{\sqrt{2}} - 1 \quad p_0 < \pi < 1
\]

References


Appendix C

Statistical and Thurstonian models for the A-not A protocol with and without sureness

Statistical and Thurstonian models for the A-not A protocol with and without sureness

Rune Haubo Bojesen Christensen a,⇑, Graham Cleaver b, Per Bruun Brockhoff a

a DTU Informatics, Technical University of Denmark, Richard Petersens Plads, Building 305, DK-2800 Kongens Lyngby, Denmark
b Unilever R&D Port Sunlight, Quarry Road East, Bebington, Wirral, CH63 3JW, UK

ABSTRACT

The A-not A protocol with sureness produce multinomial observations that are traditionally analyzed with statistical methods for contingency tables or by calculation of an R-index. In this paper it is shown that the Thurstonian model for the A-not A protocol can be written as a cumulative link model including the binormal unequal variances model. The model is extended to allow for explanatory variables and we illustrate how consumer differences can be modeled within the Thurstonian framework on a consumer study of packet soup conducted by Unilever. The extension also allows several test-product variations to be analyzed in the same model providing additional insight and reduced experimental costs. The effects of explanatory variables on the Thurstonian delta, the sensitivity (AUC), the ROC curve and the response category thresholds are discussed in detail. All statistical methods are implemented in the free R-package ordinal (http://www.cran.r-project.org/package-ordinal/).

1. Introduction

The A-not A discrimination protocol leads to count data that can be summarized in contingency tables. The traditional objective is to assess whether product differences exist. Panels of trained assessors are often used for this task, and various tests for association in contingency tables are used to analyze the data. Bi and Ennis (2001) describe various experimental designs and the appropriate statistical tests for each case.

The A-not A protocol can also be used to assess how well consumers are able to discriminate between different products. The interest is not whether a perceptible difference exists between products—often it is well known that such differences exist from the design of the products or from previous testing with trained assessors. Rather the interest is in measuring the degree to which consumers can discriminate between the products. From an analysis point of view this means that hypothesis testing is less interesting than estimation of the consumers’ ability to discriminate.

The framework of Thurstonian models provides a measure of sensory difference between products, or the respondents’ ability to discriminate between the products depending upon the view. This measure, the Thurstonian delta, is to some degree independent of the choice of discrimination protocol (e.g. Ennis, 1993). Throughout the paper we shall refer to the A-not A protocol to mean the protocols with and without sureness collectively. Whenever the distinction between the two versions is relevant, it will be mentioned. The A-not A protocol with sureness is known in the signal detection theory (SDT) literature (e.g. Green & Swets, 1966; Macmillan & Creelman, 2005) as the yes–no rating method and without sureness it is known as the yes–no task. The Thurstonian δ is known as d-prime, d', but it is the same effect measure.

The main objective of this paper is to show how consumer differences can be modeled in the A-not A protocol. It is assumed that relevant consumer characteristics have been recorded along with their ratings, and the idea is to use these characteristics to explain differences in how consumers discriminate between the products. An advantage of the approach we propose is that it is all embedded in a Thurstonian framework. In Section 2, the Thurstonian model for the A-not A protocol (with sureness) is developed and described thoroughly, and the connection to cumulative link models (CLMs) (Agresti, 2002; McCullagh, 1980) is described. The Thurstonian model for the A-not A protocol is well known and part of standard texts on SDT (Macmillan & Creelman, 2005), and the connection to cumulative link models is previously mentioned in the literature (Altham, 1980; DeCarlo, 1998; Tosteson & Begg, 1988). The connection is also important here in order to explain the Thurstonian interpretation of the more complicated models that we shall consider. Further, there does not seem to be a discussion of identifiability constraints in the Thurstonian model for the A-not A protocol in the literature to the level of detail presented in this paper. Cumulative link models can be considered as a multivariate...
extension of generalized (non-)linear models where a general regression framework is applied to categorical data. It is this property that allows the Thurstonian model for the A-not A protocol to be extended to include explanatory variables. In Section 6 it is discussed how general regression techniques as is used in ANOVA and multiple linear regression models carry over to the Thurstonian model with a few restrictions and some extensions, and how these tools can be used to provide insight into how consumers discriminate.

The regression framework makes it possible to assess consumer differences with respect to their ability to discriminate between products. For example, gender differences in the ability to discriminate can be identified, and similarly for age, lifestyle, attitudes, product usage, etc. This can be a valuable insight if the product is targeted at a particular demographic group and the relationship between the ability to detect product differences and liking may be of special interest.

Another important advantage of the modeling framework is that more accurate estimates of the Thurstonian δ are available because the estimates can be controlled for the effect of confounding variables. For instance, the values of δ for each of the genders can be estimated and controlled for the effect of e.g. age.

An extension of the A-not A protocol for replicated data is the multinomial-Diriclet model (Ennis & Bi, 1999) which, like the (corrected) beta-binomial model (Brockhoff, 2003; Ennis & Bi, 1998) for other discrimination protocols, seeks to adjust the standard error of δ for more correct hypothesis tests of δ. This can be viewed as modeling the difference between respondents with a variance parameter. The approach taken here is to model differences among respondents with explanatory variables rather than with a variance parameter. If the variation between respondents can be modeled with explanatory variables, then the model we propose is also a way of modeling replicated data. However, excess variation between respondents may still be present after controlling for the effect of explanatory variables, and formal assessment of this requires models that are beyond the scope of this paper. A small amount of un-modeled inter-respondent variability will not alter the conclusions from the models we propose appreciably.

In experiments with trained assessors as well as in experiments with consumers, it is sometimes relevant to use several similar but distinct formulations of the test product to assess whether the sensory difference between test and reference product vary between the variants of the test products. This objective can be handled by the CLM extension of the Thurstonian model for the A-not A protocol. Because this is a distinctly different objective than modeling consumer differences—variables describing the consumers are not needed, we treat it separately in Section 5.

In support of the methods in this paper, we present the R-package ordinal (Christensen, 2010), which includes implementation of all the statistical methods and models considered in this paper as well as the dataset used for the examples. The R-package is freely available for download from www.cran.r-project.org/package=ordinal. An R-script performing all the analyses is available from the online supplements or at request from the corresponding author.

The paper ends with discussion and conclusions in Section 8, and in the remainder of this section, we present a dataset which will be used for examples in Section 7.

1.1. Description of data

Unilever research conducted a discrimination study of packet soup in an A-not A protocol with sureness. Five test products were compared against a reference product by 185 respondents. Prior to testing, respondents were familiarized with the reference product. In each trial the respondent was presented with a sample and testing, respondents were familiarized with the reference product. Prior to experiment was performed, and the third group describes characteristics of the samples. The second group describes the way the experiment was performed, and the third group describes characteristics of the respondents. From the second group it is seen that four were reference products. Each respondent therefore tasted one of the test product variants twice. Due to three missing observations on three different respondents, a total of 1847 observations were obtained. The explanatory variables relevant in this study are summarized in Table 1. The variables naturally divide into three groups. The first group describes the true product status of the samples. The second group describes the way the experiment was performed, and the third group describes characteristics of the respondents. From the second group it is seen that testing was split over 2 days to minimize sensory fatigue and performed on three different locations in the Netherlands. From the third group of variables, soup type describes the type of soup most regularly consumed by the respondent and soup frequency describes the frequency with which soup is consumed by the respondent.

Some of the aims of the study were to quantify the effects of the explanatory variables and to examine whether discrimination was different for the five test products. Respondents are assumed to be approximately representative for the population of interest, so results from this study are expected to generalize well to the population of interest.

The soup dataset is available in the R-package ordinal as the object soup.

<table>
<thead>
<tr>
<th>Name</th>
<th>Code</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>prod</td>
<td>Ref, Test</td>
</tr>
<tr>
<td>Product ID</td>
<td>prod.id</td>
<td>Ref, Test1, …, Test5</td>
</tr>
<tr>
<td>Day</td>
<td>day</td>
<td>1, 2</td>
</tr>
<tr>
<td>Location</td>
<td>location</td>
<td>Region 1, Region 2, Region 3</td>
</tr>
<tr>
<td>Soup type</td>
<td>soup.type</td>
<td>Self-made, canned, dry mix</td>
</tr>
<tr>
<td>Soup frequency</td>
<td>soup.freq</td>
<td>1/week, 1–4/month, &lt;1/month</td>
</tr>
<tr>
<td>Gender</td>
<td>gender</td>
<td>Male, Female</td>
</tr>
<tr>
<td>Age</td>
<td>age</td>
<td>18–30, 31–40, 41–50, 51–65</td>
</tr>
</tbody>
</table>

Table 1 Description of variables in the soup study.

Fig. 1. Response scale used in the soup experiment.

Fig. 2. Thurstonian model with distribution of sensory intensity from reference products (solid) and test products (dashed).
2. The Thurstonian model for the A-not A protocol as a cumulative link model

The A-not A protocol (e.g., Bi & Ennis, 2001) can be used when the objective is to examine how well respondents can discriminate between two different products. The protocol typically consists of two phases: in phase one the respondents are familiarized with a reference product; the actual product testing takes place in phase two, where respondents are served a sample of either reference or test product. The respondents’ task is to try to identify which of the two types the sample is. In a generalization of the protocol, the respondents are asked to supplement their judgement (“reference” or “test”) with a sureness rating, e.g., “sure”, “unsure” and “guess” as is the case in the response scale in Fig. 1. The protocol results in responses in \( J \geq 2 \) response categories.

In the Thurstonian model, it is assumed that the sensory intensity is normally distributed (Thurstone, 1927; Torgerson, 1958) as in Fig. 2. In the A-not A protocol, it is assumed that respondents categorize each sample according to the magnitude of sensory intensity. During the familiarization process, the respondents are assumed to adopt a series of thresholds, denoted by \( \tau_i \), for the sensory intensity according to the categories on the response scale, see Fig. 2. With \( J \) response categories, \( J - 1 \) thresholds are estimable:

\[
-\infty = \tau_0 < \tau_1 < \cdots < \tau_{J-1} < \tau_J = \infty. \tag{1}
\]

The Thurstonian model for the A-not A protocol can be represented mathematically as follows: The sensory intensity, \( S \) is assumed to be a normal random variable, \( S \sim N(\mu_k, \sigma_k^2) \), where \( k = 1, 2 \) indicates the reference and test products, respectively. \( \mu_k \) and \( \sigma_k \) are the means of the distributions of sensory intensity and \( \sigma_k \) are the scales (and also the standard deviations, as the normal distribution has been assumed) of those distributions. This formulation amounts to assuming the following heterogeneous linear model for the sensory intensity of the \( k \)th sample of the \( k \)th product:

\[
S_{ik} = \mu_k + \epsilon_{ik}, \quad \epsilon_{ik} \sim N(0, \sigma_k^2). \tag{2}
\]

This model allows the location as well as the scale of the distributions of sensory intensity to differ between reference and test products. If there is no difference in the magnitude between reference and test products, the distribution of responses from reference and test samples will not differ appreciably. If there is an absolute difference, the mean sensory intensity of test products will be greater than that of reference products and \( \delta \) is positive.\(^1\) The Thurstonian \( \delta \) measures the absolute distance from reference to test products, since the response scale in Fig. 1 is uni-directional, so \( \delta \) is non-negative by definition. The actual location and orientation of the sensory distributions of reference and test products in a potentially multidimensional perceptual space is, of course, unknown.

The distributions of sensory intensity from reference and test products can be viewed as distributions of “noise” and “signal + noise”, respectively (O’Mahony, 1992). If there is some (neural and/or physical) variation associated with the signal, then the variation of the sensory intensity of test products will be larger than that of reference products and the scale of the distribution of test products will be larger than that of reference products. It may also be expected that if the reference product is familiar to the subject then the perceptual variation may be less than that for the test product.

The sensory intensity, \( S \) is not observed directly. Only the categorized version, \( Y \) is observed:

\[
Y_{jk} = j \quad \text{if} \quad \tau_{j-1} < S_{ik} \leq \tau_j.
\]

This means that the response \( Y_{jk} \) falls in category \( j \) when the sensory intensity, \( S_{ik} \) is between the thresholds \( \tau_{j-1} \) and \( \tau_j \). For instance, with the response scale in Fig. 1, a sample is categorized in “reference”, “sure” if the magnitude of the sensory intensity from the sample is less than \( \tau_1 \) and a sample is categorized as “test”, “sure” if the sensory intensity is greater than \( \tau_{J-1} \).

The response \( Y_{jk} \) can be represented by the vector \( Y_{jk} = (Y_{j1}, \ldots, Y_{jJ}) \) where \( Y_{jk} = 1 \) if \( Y_k \) falls in the \( j \)th category and zero otherwise. The probability that \( Y_{jk} \) falls in the \( j \)th category is \( P(Y_{jk} = j) = P(Y_{jk} = 1) = \pi_{jk} \) and \( Y_{jk} \) follows the multinomial distribution

\[
Y_{jk} \sim \text{Multinom}(1, \pi_k).
\]

with probability parameter vector \( \pi_k \) and index 1. The probability mass function for this distribution is the multivariate extension of the Bernoulli probability mass function:

\[
P(Y_{jk} = j) = \prod_{j=1}^{J} \pi_{jk}^{Y_{jk}}\tag{Multinom}(1, \pi_k),
\]

and the probability vector \( \pi_k \) is restricted to sum to one, \( \sum_j \pi_{jk} = 1 \) for each multinomial observation vector.

It is convenient to work with cumulative probabilities,

\[
y_{jk} = \sum_{j=1}^{J} \pi_{jk}.\]

These can be expressed as a function of the parameters describing the distributions of sensory intensity as follows (see Appendix A for details):

\[
y_{jk} = \Phi(\tau_j - \mu_k / \sigma_k),
\]

where \( \Phi \) is the standard normal cumulative distribution function.

The Thurstonian \( \delta \) is traditionally the parameter of interest defined as the distance between the distributions of sensory intensity relative to the variation of reference products; \( \delta = (\mu_2 - \mu_1) / \sigma_1 \) (Thurstone, 1927), however, none of these three parameters are directly estimable, so \( \delta \) cannot be computed in this way. The problem is that neither the absolute locations, nor the absolute scales of the distributions of sensory intensity are identifiable. By defining (without loss of generality) the relative measures,

\[
\theta = \tau_1 / \sigma_1, \quad \mu_k = \mu_k / \sigma_1 \quad \text{and} \quad \kappa_k = \sigma_k / \sigma_1,
\]

and introducing the constraints \( \mu_0 = 0 \) and \( k_1 = 1 \), the expression for the Thurstonian \( \delta \) reduces to \( \delta = \mu_2 \), which is identifiable. Other choices of constraints on \( \mu_k \) and \( \kappa_k \) are possible, but this choice provides easy inference for \( \delta \) (e.g., the standard error of \( \delta \) is directly available as the standard error of \( \mu_2 \) from software output).

An important point here is that it is neither possible to estimate nor to draw inference about \( \mu_k \) or \( \delta_k \) for any \( k \) directly. Under the Thurstonian model, the parameters, \( \tau_k, \mu_k \) and \( \sigma_k \) are real although not identifiable constants describing the distributions of sensory intensity and the respondents’ rating criteria.

Rewriting the model in terms of the relative parameters, a cumulative link model (Agresti, 2002) with a probit link function can be identified:

\[
\Phi^{-1}(y_{jk}) = \frac{\theta - \mu_k / \kappa_k}{\kappa_k}, \quad j = 1, \ldots, J - 1, \quad k = 1, 2. \tag{3}
\]

The larger statistical software packages (e.g., R (package ordinal), SAS and SPSS) can fit model (3). Model (3) is known as the unequal variances model—a special case of the binormal or normal–normal model in the SDT literature. Dorfman and Alf (1969) and Grey and Morgan (1972) early on proposed algorithms to obtain MLEs of the parameters. The identification of the binormal model as a special case of a cumulative link model means that special algorithms and special software for this model is not needed if general statistical software is available. Note that some software, assumes a plus rather than a minus in the right hand side of (3).
Inference for the relative standard deviation, $j_k$ is complicated by the restriction on the parameter space, $j_k > 0$ for all $k$ since standard deviations are positive. For instance, standard normal based Wald confidence intervals $[C_j]$ (e.g. est ± 1.96(est)) for some parameter estimate, 'est', and associated p-values are often very inaccurate. However, standard CIs and p-values are often remarkably accurate for log-$j_k$—this follows from standard theory on log-linear models (Cox & Hinkley, 1974; Pawitan, 2001). The test of no difference in scales between the distributions of reference and test products is the test of $j_k = 1$, or equivalently log-$j_k = 0$. This means that the standard two-sided normal based Wald test for log-$j_k = 0$ is appropriate and often remarkably accurate. An accurate CI for $j_k$ is obtained by computing the standard normal based Wald CI for log-$j_k$ and then transforming the confidence limits through the exponential function to obtain an appropriate CI for $j_k$. The CI for log-$j_k$ is symmetric around the MLE, log-$j_k$, whereas the CI for $j_k$ is not. More accurate CIs can be obtained from the profile likelihood (Brockhoff & Christensen, 2010; Christensen & Brockhoff, 2009; Pawitan, 2001), which is particularly relevant in small samples. Profile likelihood confidence intervals are implemented for all models considered in this paper in the R-package ordinal. More accurate significance tests (i.e. p-values) can be obtained from likelihood ratio tests as discussed in Section 2.1.

Model (3) can be generalized to accommodate a more general description of the relative location and scale. Suppose that for the $i$th observation, the location is described by $\mu_i = x_i^T\beta$ and the scale is described by log-$\sigma_i$. So a linear model is assumed for the location and a log-linear model is assumed for the scale. Here $x_i$ is a p-vector of variables for the $i$th observation with associated p-vector of location parameters $\beta$; $z_i$ is a q-vector of variables with associated q-vector of scale parameters $\zeta$. The design matrices X and Z can contain variables essential to the experiment such as those recorded in the study of packet soup. This generalization of the A-not A protocol allows the general modeling framework known from ANOVA models and multiple linear regression models as is used in the analysis of sensory profile data to carry over to this discrimination protocol—all within a Thurstonian framework and with probabilistic interpretations. The log-linear property of the scale model is chosen because it leads to easier inference as discussed above and because it ensures positivity of the scale during estimation since exp$(z_i^T\zeta)$ is positive for any $\zeta$. A general model for the $i$th observation can be written as

$$\Phi^{-1}(Y_i) = \theta - x_i^T\beta + \exp(z_i^T\zeta).$$

This model has previously been studied by Cox (1995) in the statistical literature and for ROC curve estimation in medical diagnostics by Tosteson and Begg (1988), while analysis and application of the model in a Thurstonian and sensometric context is novel. If the scale part of model (4) is restricted to unity, the model is a so-called multivariate generalized linear model (Fahrmeir & Tutz, 2001), which was considered in a sensometric context by Piepho and Kalka (2003) for sensory profile data.

2.1. Maximum likelihood estimation and model comparison

The log-likelihood function for the cumulative link model is proportional to the log of the multinomial probability mass function and is given by

$$\ell(x, y) = \sum_{i=1}^{N} y_i \log p_i,$$

where the sum is over all $N$ observations and $x$ denotes the full set of parameters. The maximum likelihood estimates, $\hat{x}$ are obtained by maximizing the log-likelihood subject to the restrictions implied on $\theta$ by Eq. (1).

Models can be compared to evaluate the evidence about a structure in the data represented by parameters. Two competing models, $m_0$ and $m_1$, where one contains one or more parameters that are not present in the larger and more general model can be compared in a likelihood ratio test. Technically speaking, the models have to be nested, such that one model is a sub-model of the other model. The likelihood ratio statistic for the comparison of $m_0$ and $m_1$, $W = -2(\ell(\hat{x}, y) - \ell(\hat{x}, y))$ asymptotically follows a $\chi^2$ distribution with degrees of freedom equal to the difference in the number of parameters between the two models. The p-value for the test measures the evidence in the data about the structure represented by the parameter(s) differing between the models.

2.2. The A-not A protocol without sureness

In the A-not A protocol without sureness, the response is observed in only $J = 2$ categories corresponding to "reference" and "test". The response is said to be Bernoulli distributed rather than multinomially and there is only one threshold, $t$ (or 0) on the magnitude of the sensory intensity. The model can therefore, in terms of the relative parameters, be written as:

$$Y_i \sim \text{Bernoulli}(\pi_i). \quad \Phi^{-1}(\pi_i) = \frac{\theta - \mu_i}{j_k}.$$

This model is not identifiable and further assumptions have to be made. Assuming that $j_k = 1$ for all $k$, i.e., that the distributions of sensory intensity differ in location, but not in scale, leads to the standard A-not A model:

$$Y_i \sim \text{Bernoulli}(\pi_i). \quad \Phi^{-1}(\pi_i) = \theta - \mu_i, \quad \mu_1 = 0.$$ (5)

Although this is a sensible a priori choice, there is technically no hindrance for assuming that the distributions of sensory intensity differ in scale and not in location. This leads to a multiplicative rather than an additive model. Model (5) was identified as a generalized linear model (GLM) (McCullagh & Nelder, 1989) with a binomial distribution and a probit link in Brockhoff and Christensen (2010).

Note the minus before $\mu_i$ in (5); most software assumes a plus. Consequently, the MLE of $\delta$ is the estimate of $\mu_j$ from the probit GLM with the sign reversed.

3. Sensitivity and area under the ROC curve (AUC)

A natural measure of sensitivity or product difference is the degree of separation of the distributions of sensory intensity for reference and test products. The degree of separation can be expressed as the probability that a sample of the reference product has a sensory intensity that is smaller than the sensory intensity of a sample of the test product. This probability is a measure of sensitivity and it can be expressed in terms of the parameters of the Thurstonian model for the A-not A protocol by simple manipulation of the normal distribution:

$$S = P(S_1 < S_2) = \Phi\left(\frac{z_2 - z_1}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right) = \Phi\left(\frac{\delta}{\sqrt{1 + k^2}}\right),$$

where $k = 1.2$ index reference and test products, respectively. The sensitivity is bounded below by the chance probability, one half, and it is bounded above at unity. This measure of sensitivity equals the area under the ROC curve (AUC) (Hanley & McNeil, 1982; Macmillan & Creelman, 2005). While it is common in sensory analysis to use "sensitivity", "ability to discriminate" and "the Thurstonian $\delta" interchangeably, this hinges on the assumption that the distributions of sensory intensity have equal scales; in simple models with no difference in scale of the distributions of sensory intensity there
is a one-to-one correspondence between the Thurstonian $\delta$ and the
sensitivity, $S$.

The sensitivity can be generalized to apply to discernible groups or
segments indexed by $g$ in a cumulative link model (4) (see
Appendix B for details):

$$S_g = \Phi\left(\frac{\delta_g}{\sqrt{\kappa^{2}_{lg} + \kappa^{2}_{kg}}}ight),$$

(7)

where $\delta_g$ is the mean difference between the distributions of
sensory intensity between reference and test products for the $g$th group
and $\kappa_{lg}$ and $\kappa_{kg}$ are the corresponding relative scales for the $g$th
group. This makes it possible to estimate and compare the sensitivity
of various consumer groups.

The R-index, equivalent to the Wilcoxon test statistic, is also an
estimator of the sensitivity, $S$ (Hanley & McNeil, 1982). This esti-
mator is consistently downwards biased under the Thurstonian model
and seems not to generalize to the situation in Eq. (7).

4. Connection to ROC curve estimation

The cumulative link models can also be used to fit ROC curves
(Green & Swets, 1966) by maximum likelihood (DeCarlo, 1998;
Tosteson & Begg, 1988). The ROC curve is a plot of the hit rate
(HR) versus the false alarm rate (FAR) on the unit square. The hit rate
is the probability of correctly rating test products as “not ref-
ence”, and the false alarm rate is the probability of falsely rating
reference products as “not reference”. Consequently the rates can
be expressed as

$$\text{FAR} = 1 - \gamma_{jk} \quad \text{for} \quad k = 1,$$

$$\text{HR} = 1 - \gamma_{jk} \quad \text{for} \quad k = 2,$$

so the (parametric) ROC curve is given by

$$\text{HR} = \Phi\left(\frac{\Phi^{-1}(\text{FAR}) - \delta_l}{\kappa_{2}}\right) \quad \text{for} \quad 0 \leq \text{FAR} \leq 1,$$

for the binormal model in Eq. (3). The more general cumulative link
model in Eq. (4) can be used to fit ROC curves for more general
designs:

$$\text{HR}_k = \Phi\left(\frac{\Phi^{-1}(\text{FAR}) - \delta_{lg} + \delta_k}{\kappa_{kg}}\right) \quad \text{for} \quad 0 \leq \text{FAR} \leq 1.$$

When the ROC curve changes, so does (most likely) also the area un-
der that curve and therefore also the sensitivity. As is clear from Eq.
(7), the sensitivity also changes when $\delta$ changes.

5. Analysis with more than one test product

The identification of the Thurstonian model for the A-not A pro-
tocol as a version of the cumulative link model allows more than
one test product formulation to be analyzed within the Thursten-
ian model. Several test products can be handled in model (3) with
$k = 1, 2, \ldots, k > 2$ and modifying the definition of $\delta$ appropri-
ately: $\delta_k = (z_k - z_{l})/\sigma_l$, which in terms of the relative parameters
reduces to $\delta_k = \mu_k$ for $k \geq 2$.

Analysis of more than one test product can be relevant when it
is desired to determine the difference between several candidate
product formulations and a reference as in the study of packet soup
described in Section 1.1. The advantage of a single experiment
including all the candidate product differences over a series of par-
allel experiments is the reduced amount of familiarization and
experimentation as well as the larger amount of information about
product differences relative to the number of parameters.

Product differences can be absent or simply be a difference be-
 tween the reference product and the entire group of test products.

With several products, there is also room for differences between
the individual test product variants. These three possible struc-
tures of product differences can be manifest in the location or in
the scale of the distributions of sensory intensity, and it may not
be the same structure that is manifest in both places. Many possi-
bile product difference structures are possible, and the correspond-
ing models can be compared by means of the likelihood. Examples
will be given in Section 7.1.

6. Assessment of the effect of explanatory variables

In this section we consider how explanatory variables can ex-
tend the Thurstonian model for the A-not A protocol. Tosteson
and Begg (1988) examined how a single explanatory variable af-
ected the shape of the ROC curve. We extend their considera-
tions and examine how an explanatory variable affects the location
and scale of the distributions of sensory intensity, how the Thurstonian
$\delta$ and the sensitivity, $S$ are affected, and how the thresholds, $\theta$
can be shifted and scaled relative to the distributions of sensory inten-
sity. This analysis makes it possible to interpret a cumulative link
model for the A-not A protocol with explanatory variables in a
Thurstonian view. We give an example of this in Section 7.2 where
we analyze the soup experiment outlined in Section 1.1.

Explanatory variables can affect the thresholds, the Thurstonian
$\delta$ and the sensitivity in various ways depending on where and how
they are included as summarized in Table 2. Consider a categorical
$X$ with two levels representing, e.g. gender, experimental session
or some other relevant explanatory variable and $\beta_1$ is the param-
eter for this variable with $i=1, 2$. If $X$ is included additively in the scale
formula corresponding to case 4 in Table 2, then the cumulative link
model for the binormal unequal variances model (3) can be written

$$\gamma_{ik} = \Phi\left(\frac{\theta_l - \mu_k}{\text{exp}(\kappa_k + \beta_k)}\right).$$

Recall that $\text{exp}(\kappa_k + \beta_k) = \text{exp}(\kappa_k) \cdot \text{exp}(\beta_k)$, so the terms in the scale
formula are multiplicative. Also recall that baseline levels are not
affected by the thresholds, the Thurstonian $\delta$ and the sensitivity, $S$ are
affected, and how the thresholds, $\theta$ can be shifted and scaled relative
to the distributions of sensory intensity. The latent distributions
then read $S_{ik} \sim N(x + \mu_k, \sigma^2 \cdot \text{exp}(\kappa_k + \beta_k))$, thus, we have

$$S_{11} \sim N(x, \sigma^2),$$

$$S_{12} \sim N(x, \sigma^2 \cdot \text{exp}(\beta_2)), $$

$$S_{12} \sim N(x, \sigma^2 \cdot \text{exp}(\kappa_2 \cdot \beta_2)),$$

$$S_{12} \sim N(x, \sigma + \mu_2, \sigma^2 \cdot \text{exp}(\kappa_2 + \beta_2)).$$

<table>
<thead>
<tr>
<th>Model</th>
<th>Locationa</th>
<th>Log scalea</th>
<th>$\phi$ Shift</th>
<th>$\theta$ Scale</th>
<th>$\delta$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>prod + X</td>
<td>prod</td>
<td>+</td>
<td></td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>prod + prod + X</td>
<td>prod + prod + X</td>
<td>+</td>
<td>+</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>prod + prod + X</td>
<td>prod + prod + X</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>prod + X</td>
<td>prod</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>prod + X</td>
<td>prod + X</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>prod + prod + X</td>
<td>prod + X</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>7</td>
<td>prod + prod + X</td>
<td>prod + X</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

* denotes inclusion and ** denotes interaction among the model terms.

Table 2: Variation of whether an explanatory variable (X) affects the sensitivity, the
Thurstonian $\delta$, the thresholds by a shift, and the thresholds by scaling depending
on how and where X has been included in the location and scale models of a
cumulative link model. The binormal unequal variances model (3) is used as a starting
point.
and $S_{22}$ are shifted $\mu_2 = \delta \sigma$-units relative to $S_{11}$ while $S_{12}$ is scaled $\exp(\kappa_2)$ relative to $S_{11}$ and $S_{22}$ is scaled $\exp(\kappa_2 + \beta_2)$ relative to $S_{11}$. Consequently $\delta$ does not differ between levels of $X$, while the sensitivity, $S$ does differ since $P(S_{11} < S_{12}) \neq P(S_{21} < S_{22})$. Observe also that if we took $\beta_2 = 0$ and estimated $\beta_1$ relative to that, the unit of measurement would be different and $\delta$ would also be different. Thus, as an estimation problem $\delta$ is only defined up to a reference category the scale of which defines the unit with which $\delta$ is measured. On the other hand $S$ is invariant to the (potentially arbitrary) choice of reference category and may therefore be a generally better effect measure.

Another way to interpret the effect of explanatory variables in numerator and denominator of the predictor is to distinguish between how the thresholds are shifted or scaled for different values of $X$. Suppose again $i = 1, 2$, and take the unequal variances model as a starting point, then including $X$ in the numerator leads to

$$\gamma_{ik} = \Phi\left(\frac{\theta_i - \mu_k - \beta_i}{\kappa_k}\right) = \Phi\left(\frac{\theta_i - \beta_i}{\kappa_k}\right),$$

where $\theta_i = \theta_0 - \beta_i$, so the thresholds are scaled for one value or level of $X$ relative to another value or level of $X$. Including $X$ in the denominator on the other hand leads to

$$\gamma_{ik} = \Phi\left(\frac{\theta_i - \mu_k}{\kappa_k} \cdot \beta_i\right) = \Phi\left(\frac{\theta_i}{\kappa_k} \cdot \beta_i\right),$$

where $\theta_i = \theta_0 / \beta_i$, so the thresholds are scaled for one value or level of $X$ relative to another value or level of $X$. In terms of the observed data, shifted thresholds means that the observed ratings tend to be centered more to the left on the response scale for one value or level of $X$ and more to the right for another value or level of $X$. Scaled thresholds implies that the extreme response categories tend to be used more frequently relative to the central categories for one level or value of $X$ than another level or value of $X$. This interpretation is not related to the underlying Thurstonian models, but it provides some insight and intuition about how the response scale is used.

Table 2 contains a summary of a number of different ways in which an explanatory variable, $X$, categorial or continuous, can be included in a cumulative link model and whether it affects the thresholds by a shift or by scaling, whether $\delta$ is affected and whether the sensitivity is affected. We use a symbolic description of model structures with separate formulas for the location and scale parts of the cumulative link model. For instance, the binormal model in Eq. (3) can be expressed as

$$\text{location} = \text{prod}, \quad \text{log scale} = \text{prod},$$

where $\text{prod}$ represents the two-levelled product factor. Several terms in the same formula can be separated by “+” (inclusion) or “:” (interaction) as exemplified in Table 2.

Table 2 shows that a shift or a scaling of the thresholds does not have to be associated with a change in $\hat{\delta}$ or sensitivity. The Thurstonian $\hat{\delta}$ is affected whenever $X$ interacts with the product indicator variable in the location, i.e., when the term $\text{prod} \cdot X$ is present in the location. The sensitivity is affected when $\hat{\delta}$ is affected or when $X$ is present in the scale model, whether it is as a main effect or in an interaction. In terms of the observed data, this implies that samples of the reference product (left side of the response scale) tend to be more separated from samples of the test product (right side of the response scale) for one value or level of $X$ relative to another value or level of $X$. A difference in sensitivity does not imply a shift or a scaling of the thresholds.

The location part of model 2 in Table 2 implies a structure with an interaction between $\text{prod}$ and $X$ without the main effect of $X$. As such it violates the so-called principle of marginality (see e.g. McCullagh & Nelder, 1989), however, the violation is apt here in light of the discussion above.

7. Examples

7.1. Example 1: Analyzing several test products

In this example, the soup data will be analyzed for differences between the test product variations. To simplify matters, the differences in experimental conditions and consumer characteristics will be ignored initially.

Model 0 in Table 3 implies a structure where the distributions of sensory intensity do not differ in either location or scale with respect to product type and only the five threshold parameters have been estimated. This is a null model against which models representing various product structures can be compared. The models in Table 3 are compared with likelihood ratio tests in Table 4. From the comparison of models 0 and 1, we see that there is strong evidence that the location of the distributions of sensory intensity for reference and test products differ.

In comparing models 1 and 2 as well as 3 and 4 we see that there is strong evidence that the locations of the test product variants differ from each other irrespective of whether the reference and test products are assumed to differ in location.

From the comparison of models 1 and 3 as well as 2 and 4 we see that there is strong evidence of a difference in scale between reference products and test products collectively. From the comparison of models 4 and 5 we see that there is practically no evidence of a difference in scale among the test product variants.

The estimates of the location and scale parameters in model 4 have been summarized in Table 5 along with standard errors and Wald based $p$-values. The estimates of the threshold parameters have been included below the table for completeness, but they are seldom of any particular interest. From Table 5 we see that the test products fall in two categories: two products have a $\delta$ of roughly 0.6 and three products have a $\delta$ of around one. The distributions of test products have a standard deviation that is around 22% larger than that of reference products.

Had five separate experiments been carried out to compare the five test product variants, then the familiarization procedure had

<table>
<thead>
<tr>
<th>Table 3 Models for several test product variants.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<p>| Table 4 Comparison of models in Table 3 with likelihood ratio tests. |
|--------------------|------------------|----------------|</p>
<table>
<thead>
<tr>
<th>Test</th>
<th>W</th>
<th>df</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 vs. 1</td>
<td>159.02</td>
<td>1</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>1 vs. 2</td>
<td>25.14</td>
<td>4</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>3 vs. 4</td>
<td>23.54</td>
<td>4</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>1 vs. 3</td>
<td>12.05</td>
<td>1</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>2 vs. 4</td>
<td>20.89</td>
<td>1</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>4 vs. 5</td>
<td>2.46</td>
<td>4</td>
<td>0.653</td>
</tr>
</tbody>
</table>
to be carried out five times rather than one, the respondents would have to taste more reference samples in order to obtain the same information to contrast reference and test products in the analysis phase, and seven parameters (five thresholds, one location and one scale parameter) would have to be fitted for each of the products, so a total of 35 parameters would have to be estimated. This should be compared to the 11 parameters used in the model chosen above. Finally, the separate experiments and analyses would not make it possible to formally assess whether location and scale differences actually differ among test product variants, as it was successfully done with these data.

7.2. Example 2: Modelling consumer differences

In this example the soup data will be analyzed with respect to differences in the experimental conditions and consumer characteristics described in Section 1.1 while estimating the Thurstonian \( \delta \) (and the sensitivity) for the five different test product formulations. A reasonable model for the soup data is:

\[
\text{location} = \text{prod.id} + \text{day} + \text{soup.type} + \text{prod:day},
\]

\[
\log \text{scale} = \text{prod}.
\]

The model is summarized in Table 6. The model includes estimates of \( \delta \) for each of the five test product formulations \( \delta_2, \ldots, \delta_6 \); observe that these estimates are different from the ones in Table 5 that were not controlled for the effect of the explanatory variables. Two additive location effects of day of experimentation and the type of soup regularly consumed are included. The parameter estimate for day indicates that the distributions of sensory intensity are shifted 0.24 scale units, i.e., standard deviations, to the left on the second day relative to the first day of experimentation. This shift affects neither the difference in location between reference and test products, nor the sensitivity, but only shifts the thresholds. However, the answers on the response scale will be further toward the left end on the second day than on the first day. Although borderline significant, the interaction between product and day indicates that \( \delta \) is around 0.26 scale units higher on the second day of experimentation than on the first day. The same additive effect is seen for all test product variants. This also means that sensitivity (AUC) is higher on the second day than on the first day, cf. Table 2. Collectively, the increase in discrimination on the second day relative to the first can be attributed to the effect that respondents were better at recognizing the reference product on the second day than on the first day.

With respect to soup type, consumers of canned soup rated further toward the left end of the response scale (”reference, sure”) than consumers of self-made soup, while consumers of dry-mix tended to rate further to the right end of the response scale (”test, sure”) than consumers of self-made soup. The likelihood ratio test of the joint effect of soup type on two degrees of freedom was significant with \( p \)-value = 0.0065. There were, however, no difference in \( \delta \) among consumers of different soup types \( p = 0.17 \). The scale of the test products is around 22% higher than the scale for reference products—an effect that was also seen in the example 7.1. Neither age, nor gender had any significant effects on location, scale or \( \delta \). Initially it seemed that the scale of the distributions of sensory intensity was smaller for respondents that consumed soup less frequently, but this effect appeared to be due to only 10 respondents that consumed soup less than once a month. For simplicity of exposition we decided to leave this structure out of the model.

To illustrate ROC curves for particular conditions, we have included ROC curves in Fig. 3 for assessment of the fifth test product on the second day (solid) and the assessment of the third test product on the first day (dotted). As an example of application of Eq. (7), the sensitivity of an assessment of the fifth test product on the second day is \( S = \Phi \left( (1.012 + 0.260) / \sqrt{1 + 1.220^2} \right) = 0.790 \). Similarly the sensitivity of an assessment of the third test product on the first day is \( S = \Phi \left( 0.471 / \sqrt{1 + 1.220^2} \right) = 0.617 \). These sensitivities correspond to the areas under the ROC curves in Fig. 3 and illustrate the relatively large differences in discriminability in these data.

### Table 5

Parameter estimates in model 4 from Table 3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Std. error</th>
<th>( p )-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta_2 )</td>
<td>0.642</td>
<td>0.091</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>( \delta_3 )</td>
<td>1.030</td>
<td>0.130</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>( \delta_4 )</td>
<td>0.601</td>
<td>0.115</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>( \delta_5 )</td>
<td>0.912</td>
<td>0.126</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>( \delta_6 )</td>
<td>1.138</td>
<td>0.135</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>( \log \kappa )</td>
<td>0.202</td>
<td>0.061</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>( \kappa_2 )</td>
<td>1.224</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The thresholds (std. error) are given by -0.899 (0.052), -0.294 (0.044), -0.08 (0.043), 0.09 (0.043) and 0.549 (0.048).

### Table 6

Cumulative probit model for the soup data.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>Estimate</th>
<th>Std. error</th>
<th>( p )-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>( \delta_2 )</td>
<td>0.508</td>
<td>0.123</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>( \delta_3 )</td>
<td>0.909</td>
<td>0.135</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>( \delta_4 )</td>
<td>0.471</td>
<td>0.131</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>( \delta_5 )</td>
<td>0.782</td>
<td>0.141</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>( \delta_6 )</td>
<td>1.012</td>
<td>0.147</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>day: 2</td>
<td></td>
<td>-0.244</td>
<td>0.079</td>
<td>0.002</td>
</tr>
<tr>
<td>soup.type: canned</td>
<td></td>
<td>-0.147</td>
<td>0.065</td>
<td>0.024</td>
</tr>
<tr>
<td>soup.type: dry-mix</td>
<td></td>
<td>0.121</td>
<td>0.083</td>
<td>0.146</td>
</tr>
<tr>
<td>prod: test, day: 2</td>
<td></td>
<td>0.260</td>
<td>0.126</td>
<td>0.039</td>
</tr>
<tr>
<td>( \log(Scale) ) prod: test</td>
<td></td>
<td>0.198</td>
<td>0.061</td>
<td>0.001</td>
</tr>
<tr>
<td>Scale</td>
<td>prod: test</td>
<td>1.220</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 3.** ROC curves for consumer sub-groups; for details see the text.

### 8. Discussion

In this paper we have considered a series of different models for binary and ordinal data. These can originate from A-not A (yes–no) tests and from rating experiments such as the A-not A with sureness experiments.

Allowing for explanatory variables in these models is a general route to the assessment of learning, and fatigue effects as well as memory effects. Assessment of these effects can be done with formal statistical models where powerful tools like likelihood ratio tests are readily applicable.
The identification of Thurstonian models as certain types of statistical regression models is also a promising route to handle replications and this paper provides a basis for understanding these more involved models. This is an area of current research.

Various familiarization schemes have been proposed for the A-not A protocol. The most distinctive difference is whether respondents are familiarized only with the reference product or also with the test product. The development of the model for the protocol is independent of whether only reference products are part of the familiarization scheme, but it may affect whether respondents actually use the discrimination strategy assumed in the model (Lee, van Hout, Hautus, & O’Mahony, 2007).

If the familiarization is with the reference product only, there is no a priori concern as to whether different sets of thresholds have been used to rate different test products. If the familiarization is with both types of products, there may be reason to assess whether the set of thresholds are different for each type of test product. This can be assessed in terms of a shift and a scaling of the thresholds in the model developed here.

### Appendix A. Formula for the cumulative probabilities

The cumulative probabilities can be written as

\[ \gamma_k = P(Y \leq j) = P(S \leq \tau_j) = P \left( Z \leq \frac{\tau_j - z_k}{\sigma_k} \right) = \Phi \left( \frac{\tau_j - z_k}{\sigma_k} \right), \]

where \( Z \) is a standard normal random variable and \( \Phi \) is the standard normal probability density function.

### Appendix B. Sensitivity by groups

The sensitivity for a group or segment, \( g \) is \( S_g = P(S_g < S_{2g}) \), where it is assumed that \( S_{1g} \sim N(\zeta_{1g}, \sigma_{1g}^2) \) and \( S_{2g} \sim N(\zeta_{2g}, \sigma_{2g}^2) \). Then,

\[ S_g = P(S_{1g} < S_{2g}) = P \left( Z < - (\zeta_{1g} - \zeta_{2g}) / (\sigma_{1g}^2 + \sigma_{2g}^2) \right) = \Phi \left( \frac{\zeta_{2g} - \zeta_{1g}}{\sqrt{\sigma_{1g}^2 + \sigma_{2g}^2}} \right), \]

where \( Z \) is a standard normal variable, \( (S_{1g} - S_{2g}) \sim N(\zeta_{1g} - \zeta_{2g}, \sigma_{1g}^2 + \sigma_{2g}^2) \), \( \zeta_k = (\zeta_{2g} - \zeta_{1g}) / \sigma_k \) and \( \sigma_{1g}^2 + \sigma_{2g}^2 = \sigma_1^2(\zeta_k^2 + \zeta_k^2) \).

### Appendix C. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.foodqual.2011.03.003.
Appendix D

Estimation of the Thurstonian model for the 2-AC protocol


A supplementary appendix describing how all examples in the paper can be executed in R using the sensR package is appended after the paper.
Estimation of the Thurstonian model for the 2-AC protocol

Rune Haubo Bojesen Christensen, Hye-Seong Lee, Per Bruun Brockhoff

Abstract

The 2-AC protocol is a 2-AFC protocol with a “no-difference” option and is technically identical to the paired preference test with a “no-preference” option. The Thurstonian model for the 2-AC protocol is parameterized by \( \delta \) and a decision parameter \( \tau \), the estimates of which can be obtained by fairly simple well-known methods. In this paper we describe how standard errors of the parameters can be obtained and how exact power computations can be performed. We also show how the Thurstonian model for the 2-AC protocol is closely related to a statistical model known as a cumulative probit model. This relationship makes it possible to extract estimates and standard errors of \( \delta \) and \( \tau \) from general statistical software, and furthermore, it makes it possible to combine standard regression modelling with the Thurstonian model for the 2-AC protocol. A model for replicated 2-AC data is proposed using cumulative link mixed models. Examples will be given throughout the paper and the methodology is implemented in the authors’ free R-packages sensR and ordinal.

1. Introduction

The 2-AC protocol is often an appealing alternative to the ordinary forced choice (2-AFC) discrimination and preference protocol (Bi, 2006; Lawless & Heymann, 1998; Macmillan & Creelman, 2005) in that it includes a no-preference or no-difference option. Often it is desirable to know if assessors or consumers really do feel that they discriminate or do have a preference—information which is disguised if a forced choice is required (Gridgeman, 1959). Further, in claim disputes, allowing for no-preference responses can be advantageous, since no-preference responses support a non-inferiority claim. However, the statistical methodology is not as well developed and well understood as the methodology for the ordinary forced choice preference or discrimination protocol (2-AFC) (Bi, Ennis, & O’Mahony, 1997; Brockhoff & Christensen, 2010). This paper seeks to rectify this shortcoming by describing the statistical methodology for estimation, power assessment and inference in the Thurstonian model for the 2-AC protocol.

Braun, Rogeaux, Schneid, O’Mahony, and Rousseau (2004) compared the 2-AFC and 2-AC protocols for discrimination of sparkling water and described a method for estimation of \( \delta \) and the decision criterion, \( \tau \) for the 2-AC model. In this paper we describe how standard errors for \( \delta \) and \( \tau \) can be found using methods from likelihood theory. We also show that the Thurstonian model for the 2-AC model is closely related to a statistical model known as the cumulative probit model (Agresti, 2002; Greene & Hensher, 2010). It follows that maximum likelihood estimates of \( \tau \) and \( \delta \) and their associated standard errors can be obtained from a cumulative probit model.

The model for the 2-AC protocol is identical to the model for the paired preference protocol (Lawless & Heymann, 1998) with the a “no-preference” option, so we will treat these together under the 2-AC protocol. Angulo and O’Mahony (2005) and Alfaro Rodriguez, O’Mahony, and Angulo (2005) considered the paired preference test with a “no-preference” option in a Thurstonian framework.

Brockhoff and Christensen (2010) described how the m-AFC, duo-trio, triangle and A-not A discrimination protocols can be identified as generalized linear models (McCullagh & Nelder, 1989) where the psychometric functions have the role of inverse link functions. An important benefit of framing the discrimination protocols in a general statistical model class is that general statistical tools can be used for discrimination data. For example the effect of explanatory variables can be assessed, e.g., the effect of gender or concentration of some additive can be quantified and tested.

Christensen, Cleaver, and Brockhoff (2011) continued this line of work and described how the A-not A protocol with sureness can be identified as a cumulative probit model. All major statistical software packages provide routines for fitting generalized linear models and cumulative probit models, so widely available software can be used to obtain maximum likelihood estimates and associated standard errors of the parameters of the 2-AC model. Further, more advanced statistical tools such as profile likelihood methods can be used to obtain better confidence intervals. See Christensen and Brockhoff (2009) and Brockhoff and Christensen (2010) for the use of profile likelihood.
methods with discrimination protocols; Pawitan (2001) provides a good statistical introduction and treatment. With the identification of the 2-AC model as a cumulative probit model in this paper, these benefits carry over to the 2-AC protocol as well.

Throughout the paper we let \( \delta \) denote the Thurstonian measure of product difference; we use hat-notation, e.g., \( \hat{\delta} \) to denote the maximum likelihood estimator or estimate of \( \delta \) a parameter. Note that \( \hat{d} \) (d-prime) is commonly used instead of \( \delta \) in the signal detection theory literature (e.g., Green & Swets, 1966; Macmillan & Creelman, 2005) while \( d \) is sometimes used in the sensometric literature (e.g., Bi et al., 1997) to denote the estimate or estimator of \( \delta \), i.e., \( \delta \) in our notation.

All methodology in this paper is implemented in the authors’ R packages sensR (Christensen & Brockhoff, 2011b) and ordinal (Christensen & Brockhoff, 2011a) freely available for the free statistical software package R (R Development Core Team, 2010). An Appendix in the on-line supplements describes how all examples can be handled in R.

The rest of the paper is organized as follows: in Section 2 we describe the decision rule for the 2-AC protocol and how parameter estimates and their standard errors can be obtained by direct methods. We also relate to classical approaches for hypothesis testing in paired comparison data with ties. In Section 3 we describe how the 2-AC model can be identified as a cumulative probit model and how parameter estimates and standard errors can be obtained through this model. We also describe the extension of the 2-AC model to a general regression framework. In Section 4 a model for replicated 2-AC data is proposed based on cumulative link mixed models, and the perspectives for assessment of heterogeneity in consumer populations with respect to preference are discussed. An example is given using data from a consumer preference test on two yoghurts with 208 consumers providing four replications each. A brief discussion and conclusions are given in Section 5. Numerical examples are given throughout the paper.

2. Direct estimation in the 2-AC model

2.1. Decision rule for the 2-AC protocol

The 2-AC protocol is a paired discrimination protocol with a “no-difference” option (Braun et al., 2004). Suppose two products, \( X \) and \( Y \) are being compared. The Thurstonian law (Thurstone, 1927a, 1927b, 1927c) assumes that the distributions of sensory intensity elicited by the products are normal, hence we assume that \( X \sim N(\mu_X, \sigma_X^2) \) and \( Y \sim N(\mu_Y, \sigma_Y^2) \). The 2-AC protocol involves a decision parameter, \( \tau \) similar in concept to the decision parameter in the same-different protocol (Christensen & Brockhoff, 2009; O’Mahony & Rousseau, 2002). The decision rule is such that an assessor will answer

1. “\( X \) is stronger than \( Y \)” if \( Y - X < -\tau \)
2. “no difference” if \( -\tau < Y - X < \tau \)
3. “\( Y \) is stronger than \( X \)” if \( Y - X > \tau \)

The distribution of differences, \( Y - X \) and where the three answers will occur are illustrated in Fig. 1.

Further, let \( n_j, j = 1,2,3 \) denote the number of observations falling in the three categories, let \( n_1 \) denote the probability that an answer falls in the \( j \)th category (cf. Fig. 1) and let \( N = \sum n_j \) denote the total number of observations.

2.2. Estimation of \( \delta \) and \( \tau \)

The spreads of \( X \) and \( Y \) are not estimable, so we assume that \( \sigma = \sigma_X = \sigma_Y \) and define \( \delta = (\mu_Y - \mu_X)/\sigma \), then \( Y - X \sim N(\delta, 2) \) which is illustrated in Fig. 1. The corresponding standardized difference distribution is shown in Fig. 2, which forms the basis for the computation of \( \tau \) and \( \delta \). It follows that

\[
\begin{align*}
\pi_1 &= P(Y - X < -\tau) = \Phi\left(-\frac{\tau - \delta}{\sqrt{2}}\right) \\
\pi_1 + \pi_2 &= P(Y - X < \tau) = \Phi\left(\frac{\tau - \delta}{\sqrt{2}}\right)
\end{align*}
\]

where \( \Phi \) is the standard normal cumulative distribution function. By equating probabilities to observed proportions, the following estimation equations are obtained:

\[
\begin{align*}
-\bar{\tau} - \bar{\delta} &= \sqrt{2}\Phi^{-1}(n_1/N) \\
\bar{\tau} - \bar{\delta} &= \sqrt{2}\Phi^{-1}(n_1 + n_2)/N
\end{align*}
\]

as is also indicated by Braun et al. (2004); see also Ennis and Ennis (2011).

2.3. Standard errors of the parameters

According to standard likelihood theory, the variance-covariance matrix of the parameters can be obtained as the inverse of the negative Hessian of the log-likelihood function evaluated at the maximum likelihood estimates. Standard errors can be obtained as the square root of the diagonal of the variance-covariance matrix. This approach to standard errors has also been taken previously in the sensometric literature, e.g., early on by Dorfman and Alf (1969) and others, and more recently by
Christensen and Brockhoff (2009) and Brockhoff and Christensen (2010). The log-likelihood function is given by
\[ \ell(x, n) = \sum l_i \log p_i, \]
where \( x = [\tau, \delta]^T \). The maximum likelihood estimates satisfy
\[ \frac{\partial \ell(x, n)}{\partial \delta} = 0 \]
and some algebra shows that the estimators in (1) satisfy this condition.

The Hessian is given by the second order derivative of the parameters evaluated at the maximum likelihood estimate, \( x \):
\[ H = \frac{\partial^2 \ell(x, n)}{\partial x \partial x^T} \mid_{x = \hat{x}} \]
The standard errors are then given by
\[ se(x) = \sqrt{\text{diag}((-H)^{-1})}. \]

The Hessian \( H \) can in practice be found by algebraic derivation or by numerical evaluation, e.g., using Richardson's extrapolation (Richardson, 1910; Richardson & Gaunt, 1927, see also Eldén, Wittmeyer-Koch, & Nielsen, 2004 for an introduction and Gilbert, 2011 for an implementation). The two-AC function in package sensR use numerical evaluation while the clm function in package ordinal evaluates the algebraic expression. In practice the numerical evaluation correctly identify all four elements of the Hessian to around six digits precision in our experience. The standard errors obtained from this method are therefore accurate beyond all practical relevance.

2.4. Example 1: Estimation of \( \delta \) and \( \tau \)
Suppose we observe \( (2, 2, 6) \) answers in the three categories. From Eq. (1) we have that \( -\tau - \delta = \sqrt{2} \hat{\delta}^{-1}(2/10) \) and \( \hat{\tau} - \delta = \sqrt{2} \hat{\delta}^{-1}(2 + 2)/10 \) from which we get \( \hat{\tau} = 0.42 \) and \( \hat{\delta} = 0.77 \). By numerical evaluation of the Hessian we obtain the standard errors: \( se(\hat{\tau}) = 0.27 \) and \( se(\hat{\delta}) = 0.54 \).

2.5. Inference in the 2-AC model
Confidence intervals for \( \delta \) can be constructed in the familiar fashion: \( CI_{\delta} : \hat{\delta} \pm z_{1-\alpha/2}se(\hat{\delta}) \), where \( z_{1-\alpha/2} = \Phi^{-1}(1 - \alpha/2) \) is the upper \( z/2 \) quantile of the standard normal distribution. This CI is based on the Wald statistic: \( w(\delta) = (\delta - \delta) / se(\hat{\delta}) \). Another statistic which generally has better properties is the likelihood root statistic (Pawitan, 2000, 2001; Boyles, 2008; Brockhoff & Christensen, 2010; Christensen & Brockhoff, 2009):
\[ r(\delta) = \text{sign}(\delta - \delta) \sqrt{2(\ell(\hat{x}, n) - \ell(\hat{\delta}, n))} \]
where \( \ell(\delta; n) = \arg\max_{\delta} \ell(\tau, \delta; n) \) is the profile likelihood for \( \delta \), i.e., the likelihood function (2) profiled over \( \tau \). The confidence interval is then given by those values of \( \delta \) that satisfy
\[ CI_{\delta} : \{ \delta : (\delta - \delta) \leq \delta \} \]
where \( \delta \) is an appropriate statistic that asymptotically follows the standard normal distribution under the null hypotheses. Using \( w(\delta) \) gives the Wald interval and using \( r(\delta) \) gives the profile likelihood interval.

The profile likelihood curve is a plot of \( \ell(\delta; n) \) as a function of \( \delta \). This curve describes the evidence or support in the data as a function of \( \delta \). The maximum likelihood estimate, \( \delta \) has the greatest support, and the further the distance from \( \delta \), the less support. For additional introduction to the use of profile likelihood curves in sensory discrimination tests see (Brockhoff & Christensen, 2010; Christensen & Brockhoff, 2009). Confidence intervals are related to the height of the relative profile likelihood curve, for Example 95% and 99% confidence limits are given as intersections with horizontal lines at 0.1465 and 0.03625 respectively—see Figs. 3 and for examples. For general \( x \) the confidence limits are given by intersections with horizontal lines at \( \exp(-\chi^2_{1,1}/2) \); this follows from the asymptotic \( \chi^2 \) distribution of the likelihood ratio statistics (cf. Pawitan, 2001). Here \( \chi^2_{1,1} \) is the \( (1 - x) \) quantile of the \( \chi^2 \) distribution with one degree of freedom.

Since the properties of the profile likelihood CIs are generally superior to those of the Wald CIs, we advise that profile likelihood CIs are used in practice. In our experience, the difference between Wald and profile likelihood intervals are, however, minor for all but very small sample sizes, say \( N < 20 \) for \( \delta \) in the 2-AC model.

The profile likelihood curve, the profile likelihood confidence intervals, and the Wald intervals are implemented in package sensR, to facilitate application of these tools.

In discrimination settings \( \delta \) is restricted to be non-negative under the Thurstonian model outlined above. On the other hand \( \delta \) may be negative in a preference setting. The consequence for confidence intervals is that they may not contain negative values in a discrimination setting, while they are not restricted in a preference setting.

In a discrimination setting one-sided hypotheses for \( \delta \) are typically most appropriate for difference and similarity tests. In a preference setting difference and similarity tests will typically be two-sided, while one-sided (non-) inferiority tests can also be relevant. The controversial issues with similarity testing (cf. Castura, 2010 and references therein) do not appear in the one-sided setting (cf. Christensen & Brockhoff, 2009), but they are relevant for the two-sided similarity tests of preference. We advise that confidence intervals are used for the assessment of difference as well as similarity whenever possible as is described by Næs, Brockhoff, and Tomic (2010, ch. 7) and proposed by Carr (1995) and MacRae (1995).

2.6. Example 2: Inference for \( \delta \)
Consider the situation in Example 1. A 95% profile likelihood confidence interval for \( \delta \) is \( (-0.27; 1.86) \), and the \( p \)-value for the one-sided discrimination test of no-difference based on the likelihood root statistic is 0.074. The relative profile likelihood is shown in Fig. 3 (solid) with an indication of 95% and 99% profile likelihood...
confidence intervals. The Wald approximation to the profile likelihood is shown in the dashed curve; the Wald approximation is quite good even in this small data set, so the Wald based p-values and confidence intervals are close to the likelihood based equivalents.

2.7. Relation to classical approaches

In the previous section we considered inference for paired preference or discrimination data with a no-preference or no-difference option via the Thurstonian model for the 2-AC protocol. In this section we discuss classical approaches to such data, but without explicit reference to the Thurstonian model. In a wider statistical and psychometric literature the data we consider are known as paired preference data with ties. The same kind of data arise from applications of the sign test when ties are allowed (Cookley & Heise, 1996; Putter, 1953; Rayner & Best, 1999).

In our notation, $n_2$ is the number of ties, while $n_1$ and $n_3$ are the number of preference responses for, say, products A and B respectively. In conventional applications of the sign test, the null hypothesis is that of no-difference, i.e., $H_0 : n_1 = n_3$ against a one- or two-sided alternative, and the hypothesis is evaluated by ignoring or discarding the ties ($n_2$) and performing a binomial test on $n_1$ versus $n_3$. Often the exact binomial test is used, but many other tests, e.g., Pearson’s $X^2$, likelihood ratio or Wald tests could be applied.

An intriguing question is if ignoring ties distorts the evidence in the data, and if it causes a loss of power. It is therefore relevant to consider tests applied to the entire data, $[n_1, n_2, n_3]$. One approach is to apply general statistical tests for contingency tables to the $1 \times 3$ table of counts, such as Pearson’s $X^2$ test or a likelihood ratio test. Another approach is to consider a test designed particularly for the sign test in the presence of ties such as Putter’s statistic (Cookley & Heise, 1996; Putter, 1955; Rayner & Best, 1999):

$$p = \frac{n_1 - n_3}{\sqrt{n_1 + n_3}},$$

which asymptotically follows a standard normal distribution under the null hypothesis of no-difference. Pearson’s $X^2$ test for the $1 \times 3$ table of counts reads

$$X^2 = \sum_{j=1}^{3} \frac{(n_j - e_j)^2}{e_j},$$

where $e_j$ are the expected counts under the null hypothesis, i.e., $e_1 = e_3 = (n_1 + n_3)/2$ and $e_2 = n_2$. The observed and expected counts are equal for the second term, so it cancels out. Therefore, in the perspective of the Pearson test, there is no information in ties and they can safely be ignored. Equivalently, ties do not contribute to the likelihood ratio test either,

$$G^2 = 2 \sum_{j=1}^{3} n_j \log \frac{n_j}{e_j},$$

so for the application of this test ties can also simply be ignored. Both $G^2$ and $X^2$ tests asymptotically follow a $\chi^2$-distribution on one degree of freedom under the null hypothesis.

Further, Pearson’s $X^2$ statistic simplifies to the square of Putter’s statistic; $X^2 = P^2$ as can be seen by expanding and simplifying Pearson’s statistic. Since the square of a standard normal variate is a $\chi^2$ variate with one degree of freedom, Pearson’s and Putter’s tests are identical and lead to identical p-values.

Other identities occur for the likelihood ratio test, since for the null hypothesis of no-difference, the likelihood ratio test in (3) is identical to the likelihood ratio and likelihood root test of $\delta$ in the Thurstonian model for the 2-AC protocol. Also, this test is identical to the likelihood ratio and likelihood root tests of $\delta$ in the Thurstonian model for the 2-AFC protocol where ties are ignored. The corresponding Wald tests are not identical.

The above mentioned identities occur only under the null hypothesis of no-difference. This means that for tests of other hypotheses such as similarity tests, the ties cannot be discarded. Similarly, the ties contribute information and cannot be ignored in the construction of confidence intervals for $\delta$.

As described by Cressie and Read (1989) and Agresti (2002) the likelihood ratio $G^2$ statistic and Pearson’s $X^2$ are very similar and usually give the same results in all but sparse tables (i.e., when the expected frequencies are larger than, say, five). The exact binomial test on the other hand has less power and almost always gives larger p-values for the no-difference test.

A classical model for paired preference data and an alternative to the Thurstonian model is the Bradley–Terry model (Bradley & Terry, 1952). The Bradley–Terry model is equivalent to the Thurstonian model for the 2-AFC protocol where a logit link is used instead of the probit link in the generalized linear model formulation of the 2-AFC model (Brockhoff & Christensen, 2010). The Bradley–Terry model is also extended to paired preferences with ties (Davidson, 1970; Rao & Kupper, 1967, but see also Agresti, 2002, Sec10.6) and is essentially a cumulative logit model similar to the cumulative probit model that we discuss in Section 3.

2.8. Power calculations

Power is the probability of rejecting the null hypothesis under a given model or alternative hypothesis, and technically it is the probability mass in the distribution of p-values below $z$. In the 2-AC model the exact p-value distribution is available since all possible outcomes can be generated, the probability of each outcome is given by the multinomial distribution, and each outcome is associated with a p-value. The number of possible outcomes depends on the sample size, $N$, and the parameters of the multinomial distribution are given by the 2-AC model parameters. The p-value for each outcome is given by the specifics of the hypothesis test and the test statistic, e.g., Wald or likelihood root.

The computationally expensive part is the computation of the p-values and the main challenge is that there may be a very large number of possible outcomes and hence a large number of p-values to compute. The number of possible outcomes depends on the sample size, $N$ and is given by the binomial coefficient; $n^* = \binom{N+3-1}{1}$, where the ‘3’ comes from the number of response categories in the 2-AC protocol. As such the number of possible outcomes, $n^*$ increase on the order of $N^2$ and the method becomes impractical for around $N = 200$, where $n^* \approx 20,000$. A small trick will, however, reduce the computational burden considerably and make the number of outcomes for which a p-value computation is required increase only linearly with $N$.

The trick is based on the fact that a large proportion of the possible outcomes will be very unlikely to occur since they are associated with very little probability mass. We therefore propose to approximate the p-value distribution by, say, 99.9999% of the distribution of outcomes by discarding consideration those outcomes that are the least likely to occur. The error in the power based on this approximation will be very small: if the p-value distribution is based on the 100(1 – $\epsilon$)% most probable outcomes, then asymptotically the error in the estimated power is at most $\epsilon$. This is practically always fulfilled for all relevant values of power

Footnote:

1. The reason that the limit is only obtained asymptotically and not exactly is due to the discreteness of the p-value distribution, which in turn derives from the discreteness of the multinomial distribution. The limit it self comes from the fact that the maximum possible error is obtained at $\alpha = 1$, where power will be $1 - \epsilon$ rather than 1 (if the p-value distribution was continuous).
(power > 50%) for \(N > 50\), and for all smaller \(N\) it is no problem to calculate the exact power. Thus, for instance, requiring that power is only estimated correctly to six significant digits, i.e. choosing \(\varepsilon = 10^{-6}\) and \(N = 1000\), a major gain in computational efficiency is obtained since \(\pi^* = 501501\), but the \(p\)-value only needs to be computed for 12684 (2.53%) of these outcomes (\(\tau = \delta = 0.2\) and \(z = 0.05\)). This takes less than half a minute on a standard laptop. Clearly, power only needs to be estimated accurately to two or three significant digits in all practical cases. The `twoACPer` function in the `sensR` package computes power for the 2-AC protocol with this method.

Power can also be estimated by using simulations to approximate the \(p\)-value distribution, but many simulations have to be run to obtain precise estimates and even then their precisions will still be approximate due to the random variations. The approach we propose is free of simulation noise and is therefore more precise and computationally more efficient. The method we propose here is in fact a general method that can be adopted to other sensory discrimination protocols. This is particularly relevant for protocols like the A-not A and same-different, where power computations are more difficult than for the simple binomial protocols; duo-trio, triangle, 2-AFC and 3-AFC, where power can be computed directly for the exact binomial test.

The power of the 2-AC model is not investigated in the literature as a function of \(\delta\) to the best knowledge of the authors although Ennis, 2010 compared the power of the 2-AC model with alternative methods. Also interesting is the dependency of the value of \(\tau\) on power. A detailed study of how power depends on sample size, parameter values and choice of test statistic is not within the scope of this paper, but would be a valuable contribution to our field.

A particular problem is associated with the Wald statistic, since in contrast to the likelihood root statistic it cannot be computed and is not defined when there is a zero-outcome in one of the cells, e.g., if all observations express a preference, since the Hessian is not positive definite so standard errors are not available. This may not be a practical problem for large sample sizes, but can be so for small sample sizes where a zero-cell outcome can be quite probable. In computing power using the Wald statistic, it needs to be decided how these zero-cell outcomes should be handled. We suggest to use the likelihood root statistic for computing power and for the reported \(p\)-values rather than the Wald statistic.

### 2.9. Example 3: Power calculation

Suppose we are planning a study in which it is expected that \(\tau = 0.5\), \(\delta = 1\), and we want to know if \(N = 20\) observations is enough to obtain reasonable power. This setting corresponds to a multinomial parameter vector of \(\pi = (0.14, 0.22, 0.64)\). The power of the two-sided preference test in this setting with \(z = 5\%\) using the likelihood root statistic is 0.778. Subject matter considerations will be needed to decide whether a larger sample size is required, or if this power is high enough.

### 3. The 2-AC model as a cumulative probit model

A cumulative link model (Agresti, 2002; McCullagh, 1980) is a model for ordered categorical (ordinal) data such as the data that is observed in the 2-AC model. In fact it turns out that the 2-AC model can be described as a cumulative link model. This gives certain advantages since general statistical software can be used to obtain maximum likelihood estimates and standard errors of \(\tau\) and \(\delta\). As we will describe, it is also possible to model and control for the effects of explanatory variables such as gender or age of the assessor and to handle various product descriptions in the same model.

In a cumulative link model the response is assumed to fall in the \(j\)th category if the latent variable \(W\) (which we can think of as representing sensory intensity) satisfies \(\theta_j - 1 < W < \theta_j\), where \(-\infty \equiv \theta_0 < \theta_1 < \theta_2 < \theta_3 \equiv \infty\) and the cumulative probabilities \(\gamma_j = \sum_{i=1}^{j} \pi_n\) are related to a linear predictor via a link function:

\[
\gamma_j = F(\hat{\theta}_j - \mu) 
\]

where \(F(\cdot)\) is the standard cumulative distribution function of the latent variable, \(W\) and \(\mu\) describes the location of the latent distribution relative to the scale. In general \(\mu\) is a linear model, for example for the \(j\)th observation we may have that \(\mu_j = \mathbf{x}_j' \beta\), where \(\mathbf{x}_j\) is a vector of predictors for the regression parameters, \(\beta\). Assuming that \(W\) is normally distributed with mean \(\mu\) leads to the assumption that \(F = \Phi\) is the standard normal cumulative distribution function, i.e. the so-called probit link.

A cumulative probit model for 2-AC data reads \(\gamma_j = \Phi(\hat{\theta}_j)\) for \(j = 1, 2, 3\). The location of the threshold parameters are illustrated in Fig. 2. By equating the expressions for \(\tau\) and \(\delta\) with \(\hat{\theta}_1\) and \(\hat{\theta}_2\), cf. Fig. 2, it follows that \(\tau\) and \(\delta\) can be found from \(\hat{\theta}_1\) and \(\hat{\theta}_2\) by:

\[
\begin{align*}
\tau &= (\hat{\theta}_2 - \hat{\theta}_1)/\sqrt{2} \\
\delta &= (\hat{\theta}_2 - \hat{\theta}_1)/\sqrt{2} 
\end{align*}
\]

(4)

The standard errors can be extracted from the variance–covariance matrix of the parameters from the fit of the cumulative probit model:

\[
\begin{align*}
\text{se}(\tau) &= \sqrt{(\text{var}(\hat{\theta}_1) + \text{var}(\hat{\theta}_2) - 2\text{cov}(\hat{\theta}_2, \hat{\theta}_1))/2} \\
\text{se}(\delta) &= \sqrt{(\text{var}(\hat{\theta}_2) + \text{var}(\hat{\theta}_1) + 2\text{cov}(\hat{\theta}_1, \hat{\theta}_2))/2} 
\end{align*}
\]

3.1. Example 4

Assuming again that we observe \((2, 2, 6)\) answers in the three categories, we obtain from a cumulative probit model the estimates: \(\hat{\theta}_1 = -0.842\) and \(\hat{\theta}_2 = -0.253\) and the following variances and covariances: \(\text{var}(\hat{\theta}_1) = 0.204\), \(\text{var}(\hat{\theta}_2) = 0.161\) and \(\text{cov}(\hat{\theta}_1, \hat{\theta}_2) = 0.111\). Inserting into Eq. (5) we get:

\[
\begin{align*}
\text{se}(\tau) &= \sqrt{(0.204 + 0.161 - 2 \cdot 0.111)/2} = 0.267 \\
\text{se}(\delta) &= \sqrt{(0.204 + 0.161 + 2 \cdot 0.111)/2} = 0.542, 
\end{align*}
\]

as in Example 1.

3.2. A general regression model for \(\delta\)

The relation between the 2-AC model and the cumulative probit model makes it possible to control for and assess the effect of explanatory variables much in the same way Brockhoff and Christensen, 2010 extended m-AFC, duo-trio, triangle and A-not A methods, and similarly how Christensen et al., 2011 extended the A-not A model with sureness. Suppose for example that it is expected that discrimination differs between genders. A cumulative probit model that models this difference reads:

\[
\gamma_j = \Phi(\hat{\theta}_j - \chi x),
\]

where \(i = 1, 2\) for females and males respectively and \(x\) is a dummy variable. If we choose females as the reference category so \(x_1 = 1\) for males and zero otherwise, then \(x\) is the shift of the thresholds, \(\theta_j\) for males relative to the thresholds for the females. The \(\tau\) and \(\delta\) estimates for females are given by Eq. (4), but those for males are given by:
The result is that $\tau$ remains unchanged because $x$ cancels out while $\delta$ for males is shifted by $\sqrt{2}$ relative to $\delta$ for females. A cumulative probit model is therefore a regression model for $\delta$, while $\tau$ is assumed constant across explanatory variables.

A test of the assumption of constant $\tau$ can be framed as a model comparison where the cumulative probit model is compared to a model in which both $\tau$ and $\delta$ are estimated for every combination of the explanatory variables—the latter model being an aggregation of 2-AC models fitted to these combinations. Natural choices of test statistics are likelihood ratio and Pearson's $X^2$ statistics, both being compared to a $\chi^2$ distribution with degrees of freedom equal to the difference in the number of parameters. Further details are not given here, but an example of the likelihood ratio test is given in Example 5 below and an application of Pearson's $X^2$ test is given in the online supplements.

A test for gender differences in $\delta$ is the test of $x=0$ for which standard likelihood ratio tests or Wald tests can be applied. The standard error of the $\delta$ difference between genders is given by:

$$se(\delta) = \sqrt{\frac{2}{se(x)}}$$

and $se(\delta)$ is available from the fit of a cumulative probit model. The Wald test for $x$ reported by the cumulative probit model is identical to the test of $\delta/\sqrt{2}$ since the test statistic, $z$, is given by

$$z = \frac{x/\sqrt{2}}{se(x/\sqrt{2})} = \frac{x}{se(x)} \cdot \frac{\sqrt{2}}{\sqrt{2}} = \frac{x}{se(x)}$$

The cumulative probit model for the 2-AC model extends easily to a general regression model for $\delta$ differences:

$$\gamma_0 = \Phi(h_1 - x_i^b \beta)$$

where $x_i$ is a vector of explanatory variables for the regression coefficients $\beta$. The shift in $\delta$ for the $i$th observation (relative to the chosen baseline) is then given by $x_i^b \beta/\sqrt{2}$.

### 3.3. Profile likelihood and confidence intervals

Confidence intervals for $x$ can be constructed by profile likelihood (Boyles, 2008) such as implemented in the ordinal package (Christensen & Brockhoff, 2011a) or by traditional Wald intervals.

#### 3.4. Example 5

Assume that a study is performed where gender differences in discrimination ability is of interest. Suppose that (20, 20, 60) is observed for women and (10, 20, 70) is observed for men. The output from a cumulative probit model contains: $\hat{h}_1 = -0.889$, $\hat{h}_2 = -0.228$, $\hat{\delta}_{\text{gender}} = 0.32$ with standard error $se(\hat{\delta}_{\text{gender}}) = 0.174$. Using the formulas in Eq. (4), we find that $\tau = 0.467$, $\hat{\delta}_{\text{females}} = 0.790$ and $\hat{\delta}_{\text{males}} = \hat{\delta}_{\text{females}} + \hat{\delta}_{\text{gender}} \sqrt{2} = 1.24$. The Wald test for $\hat{\delta}_{\text{gender}}$ gives $p = 0.066$ while the likelihood ratio test yields $p = 0.065$. Consequently the gender difference is marginally significant. A profile likelihood 95% confidence interval for the gender difference in $\delta$, $\hat{\delta}_{\text{gender}} \sqrt{2}$ is $(0.029, 0.937)$. The gender difference may therefore with reasonable confidence be as large as one $\delta$-unit or practically non-existing.

A test of the constancy of $\tau$ across genders can be constructed by comparing our model (with three parameters) to a model that fits one $\tau$ and one $\delta$ for each gender (four parameters). The likelihood ratio test statistic is twice the difference in likelihood for the two models, where the likelihood for the second model is given as the sum of likelihoods for two 2-AC models; one for each gender. The log-likelihood for the cumulative probit model allowing for a gender difference in $\delta$ is $-175.59$, while the log-likelihoods for the 2-AC models for women and men are $-95.03$ and $-80.18$ respectively. The likelihood ratio test statistic for the test of constancy of $\tau$ across genders is therefore $LR = \chi^2 (175.59 - (95.03 - 80.18)) = 0.7683$, which, under the null hypothesis of no $\tau$ difference, is asymptotically distributed as a $\chi^2$-distribution with one degree of freedom corresponding to the difference in the number of parameters. The associated $p$-value is 0.38, which does not provide any evidence of a difference in $\tau$ across genders.

### 4. Replicated 2-AC data

Replicated 2-AC data are observations resulting from a 2-AC protocol where each assessor is allowed to perform more than one assessment. Cumulative link models extend naturally to a replicated setting by allowing for random effects for assessors; the resulting model is known as a cumulative link mixed (effects) model, since the model allows for both fixed effects and random effects. We refer to Christensen and Brockhoff (2011a) for a description of this model in a sensometric context and for further references. In this section we describe how cumulative link mixed models can be used to fit and analyze replicated 2-AC data. A clear advantage is that the model we propose can be fitted and analyzed with standard statistical software, for example with the `clmm` function in the first author's ordinal R-package (Christensen & Brockhoff, 2011a).

The (corrected) beta-binomial models (Bi & Ennis, 1998; Brockhoff, 2003; Ennis & Bi, 1998; Meyners, 2007) are classical approaches when dealing with replicated binomial data in sensometrics. A generalized linear mixed model for instance considered by Brockhoff (2003) is more closely the equivalent of cumulative link mixed (effects) models for binomial data. A Dirichlet-Multinomial model suggested by Ennis and Bi (1999) is an extension of the beta-binomial model for multinomial data, and this model could also be considered for replicated 2-AC data. However, this model does not have the direct link to the Thurstonian model for the 2-AC protocol from which the cumulative probit link model stems.

A cumulative link mixed model that allows for a general regression model of explanatory variables as well as random effects for assessors reads

$$\gamma_i = \Phi(h_i - x_i^b \beta - b_i), \quad b_i \sim N(0, \sigma_b^2)$$

where $k = 1, \ldots, m$ assessors each perform $i = 1, \ldots, n$ assessments, $x_i^b$ is the value of the explanatory variables on the $i$th observation (relative to the chosen baseline) is then given by $x_i^b \beta/\sqrt{2}$.
positive \( \delta \) are expected. While a negative \( \delta \) is not statistically impossible in the 2-AC model, as it is in triangle and duo-trio experiments adequately designed and conducted (Kunert, 2001; Kunert & Meyners, 1999), it might be unexpected and would require that an assessor has his perceptual or sensory dimension reversed. If, for example, the question is "which sample is perceived as the most salty", then we know which product contains the most salt, and a negative \( \delta \) is realized if the less salty product is truly perceived as more salty than the product with the highest salt concentration. Even though not statistically impossible, it might be sensorically and practically unlikely that this happens—this will rest on subject matter considerations.

In light of these reflections we argue that the proposed model can be adequate also for discrimination tests. When the mean \( \delta \) is not small and/or the variation is moderate or small, the distribution of \( \delta \) will have most of its mass above zero, so there is no practical problem with the model. When the mean \( \delta \) is close to zero, there will be some probability mass below zero which might render the model inappropriate. In any case it will be possible with the proposed model to quantify the extent to which \( \delta \) varies in the study population.

4.1. Example 6

In this example we will consider data from Kim and Lee (2010) where 208 consumers performed a combination of duo-trio discrimination tests and 2-AC preference tests on a pair of commercial yoghurts (denoted A and B). We will focus on the 2-AC data where each consumer conducted four tests. In addition to the response ("prefer A", "no preference" and "prefer B"), it was also recorded whether A or B was the reference sample in the preceding duo-trio test (we denote this variable reference), and whether the duo-trio test was of the DTM or DTFR type (we denote this variable protocol). In the DTM test the reference sample is presented in between the two test samples, and in DTFR reference samples are presented before as well as after the two test samples (Rousseau, Stroh, & O'Mahony, 2002, Lee & Kim, 2008 and Kim, Lee, & Lee, 2010). In addition, consumers were divided into eight groups and each group performed the tests in a particular order. The data set is balanced with respect to all variables and with no missing values; for further details about the experimental design see Kim and Lee, 2010.

All consumers performed the preference task at each combination of the reference and protocol settings resulting in 208 \( \times \) 2 \( = \) 832 preference observations. Marginally, product A was preferred in 57% of the cases, "no preference" was expressed in 9% and preference for product B was expressed in 34% of the cases. Ordering the response options as "prefer A" < "no preference" < "prefer B" leads to a negative mean estimate of \( \delta \): \( \hat{\delta} = -0.429, \text{se}(\hat{\delta}) = 0.0599 \).

Of interest here is first of all whether it is possible to identify structured differences in preference relating to the levels of protocol and reference variables, and second, whether there is an additional variation in preference (\( \delta \)) among consumers. Lastly, a clear and significant effect of the group variable will indicate that the order in which the experiment is conducted will affect preference, so we can use a test of this variable as a test for unwanted effects of the experimental design. The Laplace approximation was used for intermediate model fits during model reduction and ten-point adaptive Gauss-Hermite quadrature was used for the model presented in Table 1 (cf. Christensen & Brockhoff, 2011a). This usage of the computational methods follows the advice in Joe (2008) and the methods are available options in the \texttt{nlme} function in package \texttt{ordinal}.

In the application of model (6) we found that there was no hint of a group effect while controlling for the remaining variables (\( G^2 = 4.88, \text{df} = 7 \) and \( p\text{-value} = 0.66 \)) and therefore not indicating that unwanted effects of the experimental design are influencing the conclusions from our analysis. Here \( G^2 \) denotes the likelihood ratio test statistic. Similarly, we did not find any effect of protocol while controlling for the remaining variables (\( G^2 = 1.12, \text{df} = 1 \) and \( p\text{-value} = 0.30 \)) indicating that whether the DTM or DTFR duo-trio protocol preceded the preference test did not appreciably affect the preference task. This is supported by the estimate of the difference in \( \hat{\delta} (0.105) \) and 95% confidence interval \([-0.088, 0.298]\) indicating that any possible \( \delta \) difference is small. On the other hand we found that there is a clear difference as to whether the A product or the B product appeared as the reference sample in the duo-trio test preceding the preference test (\( G^2 = 16.4, \text{df} = 1 \) and \( p\text{-value} < 0.0001 \)). A summary of the resulting model allowing for different \( \delta \) estimates for A and B reference products appear in Table 1.

The model describes that A products are more preferred when A is also the reference product in the duo-trio test preceding the preference test. This can be understood as a consequence of the mechanism that consumers tend to prefer the product to which they have been most exposed. In the DTM version of the duo-trio protocol each consumer is exposed to the reference product twice and the test product once, while in the DTFR version of the duo-trio protocol each consumer is exposed to the reference product three times and the test product once. In this light it could be hypothesized that the higher preference for the reference product would be more pronounced for the DTFR version than for the DTM version of the duo-trio protocol. This hypothesis can be tested by testing the interaction between the protocol and reference variables. There is, however, no appreciable evidence in the data about such a structure (\( G^2 = 2.28, \text{df} = 2 \) and \( p\text{-value} = 0.32 \)). An important advantage of embedding the 2-AC protocol in the framework of cumulative link models is the ease with which such hypotheses can be tested; it is just a standard test in a regression model. Without the model framework tests of these hypotheses are generally unavailable.

The variation in \( \hat{\delta} \) among consumers is considerable compared to the size of the \( \delta \) estimates (cf. Table 1): approximately 95% of the population will have a preference within \( \pm 2 \delta \approx \pm 3.3 \), so preference will span the whole range from consistently preferring B in this consumer population—this essentially holds for both types of reference products since the difference due to reference product is small compared to the variation in preference among consumers.

The probability that a consumer will prefer yoghurt A, have no preference, or prefer yoghurt B is illustrated in Fig. 4 based on the model in Table 1. The top panel describes the probabilities for an average consumer while the two lower panels describe the probabilities for consumers in the tails of the distribution—here the 5th percentile and 95th percentile. An average consumer has \( b = 0 \), while \( b = 1.64 \delta \) for the extreme consumers.2 While there is a moderate difference between the rating probabilities for average consumers, this difference vanishes for the more extreme

\( ^2 \text{ 1.64} = \Phi^{-1}(0.95) \) is the 95th percentile of the standard normal distribution.

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>Lower</th>
<th>Upper</th>
<th>( p\text{-Value} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau )</td>
<td>0.259</td>
<td>0.029</td>
<td>0.202</td>
<td>0.316</td>
</tr>
<tr>
<td>( \sigma_{r} )</td>
<td>-0.941</td>
<td>0.160</td>
<td>-1.254</td>
<td>-0.628</td>
</tr>
<tr>
<td>( \sigma_{b} )</td>
<td>-0.367</td>
<td>0.154</td>
<td>-0.668</td>
<td>-0.066</td>
</tr>
<tr>
<td>( \sigma_{\delta} )</td>
<td>1.654</td>
<td></td>
<td>1.362</td>
<td>2.001</td>
</tr>
<tr>
<td>Log-likelihood</td>
<td>-668.9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Summary of a cumulative link mixed model (6) fit to the yoghurt preference data considered in Example 6 including 95% confidence limits. Confidence limits and \( p\text{-values} \) are based on the Wald statistic except for \( \sigma_{\delta} \) for which they are based on the likelihood root statistic.
consumers. Clearly the most dominant effect is the difference among consumers rather than the difference caused by the reference consumer. This illustrates the difference between population-average and subject-specific parameters (cf. Christensen & Brockhoff, 2011a); the parameters of the mixed effects model apply at the subject or consumer level, whereas the parameters of the fixed effect model apply at the population level.

Similarly, the uncertainty as measured by the standard errors is larger in the mixed effects model. Notably the significance of $\delta_{\text{ref}}$ is deceptively much stronger in the fixed effects model than in the proper mixed effects model. Most importantly this analysis is sensible about the largest and most significant effect in these data: namely the variation in preference among consumers.

### 4.2. Assessing heterogeneity in consumer populations

Caution needs to be taken when the 2-AC model is used for studying preference at the population level; the average preference in the population may be zero even if there is a perceptible sensory difference between the products.

We cannot be sure if consumers will prefer one product over the other even when consumers can discriminate between the two products based on their sensory properties; it is possible that the two products are perceived differently but still preferred equally. It is also possible that the consumer population is heterogeneous with respect to preference in such a way that preferences for the two products balance out in the population. In the latter situation it is not possible to identify or understand the source of heterogeneity in preference without replications.

If two products are not discriminated, assessors are expected to respond primarily to the "no-preference" option. However, it has been observed in previous literature that subjects tend to report performance or a bias performance.

Importantly, biased preference responses could also occur when comparing two sensorically clearly distinguishable products. Without replications it would be difficult to assess if preference responses occur due to the demanding characteristics of the test method. Importantly, biased preference responses could also occur when comparing two sensorically clearly distinguishable products. Without replications it would be difficult to assess if preference responses occur due to the demanding characteristics of the test method. Importantly, biased preference responses could also occur when comparing two sensorically clearly distinguishable products. Without replications it would be difficult to assess if preference responses occur due to the demanding characteristics of the test method. Importantly, biased preference responses could also occur when comparing two sensorically clearly distinguishable products. Without replications it would be difficult to assess if preference responses occur due to the demanding characteristics of the test method. Importantly, biased preference responses could also occur when comparing two sensorically clearly distinguishable products. Without replications it would be difficult to assess if preference responses occur due to the demanding characteristics of the test method.
This is important insight, since for instance, if preference is differentiated by age, then different products should be targeted at different age segments. This is less relevant if preference heterogeneity is not explained by any observed variable.

5. Discussion and conclusions

We have described how estimates and standard errors can be obtained for δ and τ, the parameters of the Thurstonian model for the 2-AC protocol. We have also clarified the practical and scientifically relevant connection between the Thurstonian model for the 2-AC protocol and cumulative probit models. An immediate advantage of this identification is that estimates and standard errors of δ and τ are directly available from general statistical software, e.g. the freely available, internationally recognized and quickly expanding (Fox, 2008) R-program (R Development Core Team, 2010) via the authors’ R-package ordinal (Christensen & Brockhoff, 2011a). We have shown how cumulative probit models can be used in the modern statistical modeling framework well-known from ANOVA and multiple linear regression models for normally distributed data with discrimination and preference testing in the 2-AC model. This continues the line of work initiated by Brockhoff and Christensen (2010) where several common discrimination protocols (m-AFC, duo-trio, triangle and A-not A) were identified as generalized linear models and continued by Christensen et al. (2011) where the A-not A protocol with sureness was identified as a cumulative probit model. We also proposed a model for replicated 2-AC data that builds on cumulative link mixed models; a class of models that are useful for analyzing categorical ratings data from sensory experiments as shown by Christensen and Brockhoff (2011a). We believe these extensions have considerable potential for future applications of discrimination testing in industry and research.

Acknowledgments

The authors thank two reviewers for comments and suggestions that helped improve the paper. The first author also thanks Roy Roberts for motivating the paper and for stimulating discussions.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.foodqual.2011.10.005.

References


Richardson, L. F. (1910). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London, Series A, Containing Papers of a Mathematical or Physical Character*, 210, 307–357.


This document describes how the examples in “Estimation of the Thurstonian model for the 2-AC protocol” (Christensen et al., 2011) can be executed in the free statistical software R (R Development Core Team, 2011) using the free R packages sensR and ordinal (Christensen, 2011; Christensen and Brockhoff, 2011) developed by the authors.

1 Example 1: Estimation of \(d'\) and \(\tau\)

It is assumed that we have \(n = (2, 2, 6)\) observations in the three categories. We may estimate \(\tau\), \(d'\) and their standard errors with the \texttt{twoAC} function in the \texttt{sensR} package:

\begin{verbatim}
> library(sensR)
> fit <- twoAC(c(2, 2, 6))
> fit

Results for the 2-AC protocol with data c(2, 2, 6):

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tau)</td>
<td>0.4159726</td>
</tr>
<tr>
<td>(d')</td>
<td>0.7742595</td>
</tr>
</tbody>
</table>

Two-sided 95% confidence interval for \(d'\)-prime based on the likelihood root statistic:

<table>
<thead>
<tr>
<th></th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d')</td>
<td>-0.2709832</td>
<td>1.859251</td>
</tr>
</tbody>
</table>

Significance test:

Likelihood root statistic = 1.446718 p-value = 0.14798

Alternative hypothesis: \(d'\)-prime is different from 0

Alternatively we may compute \(\tau\) and \(d'\) manually:

\begin{verbatim}
> n <- c(2, 2, 6)
> gamma <- cumsum(n/sum(n))
> z <- qnorm(gamma)[-3]
\end{verbatim}
2 Example 2: Inference for d-prime

The likelihood based confidence intervals and the one-sided discrimination significance test where the null hypothesis is “no sensory difference”, i.e., \( d'_0 = 0 \) using the likelihood root statistic are immediately available using the \texttt{twoAC} function:

```r
> twoAC(c(2, 2, 6), d.prime0 = 0, conf.level = 0.95, statistic = "likelihood",
    alternative = "greater")
```

Results for the 2-AC protocol with data c(2, 2, 6):

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau )</td>
<td>0.4159726</td>
<td>0.2674304</td>
</tr>
<tr>
<td>( d' )</td>
<td>0.7742595</td>
<td>0.5416717</td>
</tr>
</tbody>
</table>

Two-sided 95% confidence interval for \( d' \) based on the likelihood root statistic:

<table>
<thead>
<tr>
<th></th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d' )</td>
<td>-0.2709832</td>
<td>1.859251</td>
</tr>
</tbody>
</table>

Significance test:

Likelihood root statistic = 1.446718 p-value = 0.073988
Alternative hypothesis: \( d' \) is greater than 0

The relative profile likelihood and Wald approximation can be obtained with:

```r
> pr <- profile(fit)
> plot(pr)
> z <- pr$d.prime$d.prime
> w <- (coef(fit)[2, 1] - z)/coef(fit)[2, 2]
> lines(z, exp(-w^2/2), lty = 2)
```

3 Example 3: Power calculations

The function \texttt{twoACpwr} computes the power for the 2-AC protocol and a significance test of the users choice. The power assuming \( \tau = 0.5, d' = 1 \), the sample size is \( N = 20 \) for a two-sided preference test with \( \alpha = 0.5 \) is found with:

```r
> twoACpwr(tau = 0.5, d.prime = 1, size = 20, d.prime0 = 0, alpha = 0.05,
    alternative = "two.sided", tol = 1e-05)
```

<table>
<thead>
<tr>
<th></th>
<th>p.1</th>
<th>p.2</th>
<th>p.3</th>
<th>p.4</th>
<th>p.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.04960103</td>
<td>137</td>
<td>0.2174</td>
<td>0.1444</td>
<td>0.6382</td>
</tr>
</tbody>
</table>
Figure 1: Relative profile likelihood curve (solid) and Wald approximation (dashed) for the data in example 1 and 2. The horizontal lines indicate 95% and 99% confidence intervals based on the likelihood function and the Wald approximation.

Apart from the power, we are told that the actual size of the test, \( \alpha \) is close to the nominal 5%. The reason that the two differ is due to the discreteness of the observations, and hence the test statistic. We are also told that with \( N = 20 \) there are 231 possible outcomes of the 2-AC protocol. In computing the power 94 of these are discarded and power is computed based on the remaining 137 samples. The fraction of samples that are discarded is determined by the tol parameter. If we set this to zero, then no samples are discarded, however, the increase in precision is irrelevant:

\[
> \text{twoACpwr}(\tau = 0.5, d.\text{prime} = 1, \text{size} = 20, d.\text{prime0} = 0, \alpha = 0.05, \\
\quad \text{alternative} = "\text{two.sided}", \text{tol} = 0)
\]

<table>
<thead>
<tr>
<th>power</th>
<th>actual.\alpha</th>
<th>samples discarded</th>
<th>kept</th>
<th>p.1</th>
<th>p.2</th>
<th>p.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7776788</td>
<td>231</td>
<td>0</td>
<td>231</td>
<td>0.1444</td>
<td>0.2174</td>
</tr>
</tbody>
</table>

The last three numbers in the output are the cell probabilities, so with \( \tau = 0.5 \) and \( d' = 1 \) we should, for example, expect around 22% of the answers in the “no difference” or “no preference” category.

4 Example 4: Estimation and standard errors via cumulative probit models

Estimates of \( \tau \) and \( d' \) and their standard errors can be obtained from a cumulative probit model. We begin by defining the three leveled response variable resp and fitting a cumulative link model (CLM) using the function clm in package ordinal with weights equal to the observed counts and a probit link. Standard output contains the following coefficient table:
```r
> response <- gl(3, 1)
> fit.clm <- clm(response ~ 1, weights = c(2, 2, 6), link = "probit")
> (tab <- coef(summary(fit.clm)))

|   | Estimate | Std. Error | z value | Pr(>|z|) |
|---|----------|------------|---------|----------|
| 1 | -0.8416212 | 0.4518154 | -1.8627547 | 0.06249679 |
| 2 | -0.2533471 | 0.4009896 | -0.6318047 | 0.52751451 |

The $\tau$ and $d'$ estimates are obtained by:

```r
t > theta <- tab[, 1]
> (tau <- (theta[2] - theta[1])/sqrt(2))
> 2|3
> 0.4159726
```

```r
t > (d.prime <- (-theta[2] - theta[1])/sqrt(2))
> 2|3
> 0.7742595
```

The variance-covariance matrix of the parameters can be extracted by the `vcov` method and the standard errors computed via the provided formulas:

```r
> VCOV <- vcov(fit.clm)
> (se.tau <- sqrt((VCOV[1, 1] + VCOV[2, 2] - 2 * VCOV[2, 1])/2))
1 [1] 0.2674311
> (se.d.prime <- sqrt((VCOV[1, 1] + VCOV[2, 2] + 2 * VCOV[2, 1])/2))
1 [1] 0.5416737
```

Observe how these estimates and standard errors are identical to those in Example 1.

We could also have used the `clm2twoAC` function from package `sensR` which extract estimates and standard errors from a `clm` model fit object:

```r
> clm2twoAC(fit.clm)

|   | Estimate | Std. Error | z value | Pr(>|z|) |
|---|----------|------------|---------|----------|
| tau | 0.4159726 | 0.2674311 | 1.555439 | 0.11984 |
| d-prime | 0.7742595 | 0.5416737 | 1.429384 | 0.15289 |

5 Example 5: A regression model for $d'$

Assume that a study is performed and gender differences in the discrimination ability is of interest. Suppose that (20, 20, 60) is observed for women and (10, 20, 70) is observed for men. The standard output from a cumulative probit model contains the coefficient table:

```r
> n.women <- c(2, 2, 6) * 10
> n.men <- c(1, 2, 7) * 10
> wt <- c(n.women, n.men)
> response <- gl(3, 1, length = 6)
> gender <- gl(2, 3, labels = c("women", "men"))
> fm2 <- clm(response ~ gender, weights = wt, link = "probit")
> (tab2 <- coef(summary(fm2)))
```
```
The estimate of \( \tau \) (assumed constant between genders) and the gender specific estimates of \( d' \) can be extracted by:

\[
\begin{align*}
\theta & \leftarrow \text{fm2$alpha} \\
(\tau) & \leftarrow (\theta[2] - \theta[1])/\sqrt{2} \\
(d.\text{women}) & \leftarrow (-\theta[2] - \theta[1])/\sqrt{2} \\
(d.\text{men}) & \leftarrow (d.\text{women} + \text{fm2$beta} * \sqrt{2})
\end{align*}
\]

Again we could use the \texttt{clm2twoAC} function to get the coefficient table for the 2-AC model from the CLM-model:

\[
\begin{align*}
\text{Estimate Std. Error z value Pr(>|z|)} \\
\tau & 0.4670001 0.0670001 4.970135 3.1664e-12 \\
d-prime & 0.7898785 0.1702127 4.640538 3.4750e-06 \\
gendermen & 0.4533125 0.2462774 1.840658 6.5672e-02
\end{align*}
\]

Observe that \( d' \) for women is given along with the difference in \( d' \) for men and women rather than \( d' \) for each of the genders.

The Wald test for gender differences is directly available from the coefficient table with \( p = 0.0657 \). The corresponding likelihood ratio test can be obtained by:

\[
\begin{align*}
\text{fm3} & \leftarrow \text{update(fm2, }, ~\text{ gender}) \\
\text{anova(fm2, fm3)}
\end{align*}
\]

Likelihood ratio tests of cumulative link models:

\[
\begin{align*}
\text{formula:} & \quad \text{link: threshold:} \\
\text{fm3} & \quad \text{probit flexible} \\
\text{fm2} & \quad \text{gender probit flexible}
\end{align*}
\]

\[
\begin{align*}
\text{no.par} & \text{ AIC logLik LR.stat df Pr(>Chisq)} \\
\text{fm3} & 2 358.59 -177.29 \\
\text{fm2} & 3 357.19 -175.59 3.3997 3.840658 6.567176e-02
\end{align*}
\]

which is slightly closer to significance. The 95% profile likelihood confidence interval for the difference between men and women on the \( d' \)-scale is:

\[
\begin{align*}
\text{confint(fm2) * sqrt(2)}
\end{align*}
\]

5
The likelihood ratio test for the assumption of constant $\tau$ is computed in the following. The likelihood ratio statistic and associated $p$-value are

```r
> logLik(fm2)
'log Lik.' -175.593 (df=3)
> tw <- twoAC(n.women)
> tm <- twoAC(n.men)
> (LR <- 2 * (tw$logLik + tm$logLik - fm2$logLik))
[1] 0.7682623
> pchisq(LR, 1, lower.tail = FALSE)
[1] 0.3807552
```

The Pearson $X^2$ test of the same hypothesis is given by

```r
> freq <- matrix(fitted(fm2), nrow = 2, byrow = TRUE) * 100
> Obs <- matrix(wt, nrow = 2, byrow = TRUE)
> (X2 <- sum((Obs - freq)^2/freq))
[1] 0.7657565
> pchisq(X2, df = 1, lower.tail = FALSE)
[1] 0.381533
```

so the Pearson and likelihood ratio tests are very closely equivalent as it is so often the case.

### 6 Regression model for replicated 2-AC data

The data used in example 6 are not directly available at the time of writing. Assume however, that data are available in the R `data.frame` `repData`. Then the cumulative probit mixed model where preference depends on reference and consumers (`cons`) are random can be fitted with:

```r
> fm3.agq <- clmm2(preference ~ reference, random = cons, nAGQ = 10,
data = repData, link = "probit", Hess = TRUE)
> summary(fm3.agq)
```

Cumulative Link Mixed Model fitted with the adaptive Gauss-Hermite quadrature approximation with 10 quadrature points

**Call:**

clmm2(location = preference ~ reference, random = cons, data = repData,
      Hess = TRUE, link = "probit", nAGQ = 10)

**Random effects:**

<table>
<thead>
<tr>
<th>Var</th>
<th>Std.Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>cons</td>
<td>1.367074 1.169219</td>
</tr>
</tbody>
</table>
Location coefficients:

|            | Estimate | Std. Error | z value | Pr(>|z|) |
|------------|----------|------------|---------|----------|
| referenceB| 0.4059   | 0.1002     | 4.0519  | 5.0801e-05 |

No scale coefficients

Threshold coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>N</td>
<td>0.4823</td>
<td>0.1130</td>
</tr>
<tr>
<td>N</td>
<td>B</td>
<td>0.8488</td>
<td>0.1166</td>
</tr>
</tbody>
</table>

log-likelihood: -668.9122
AIC: 1345.824
Condition number of Hessian: 36.7977

Here we asked for the accurate 10-point adaptive Gauss-Hermite quadrature approximation. The 2-AC estimates are available using the `clm2twoAC` function:

```r
> clm2twoAC(fm3.agq)
```

|            | Estimate   | Std. Error   | z value    | Pr(>|z|) |
|------------|------------|--------------|------------|----------|
| tau        | 0.2592192  | 0.02902566   | 8.930689   | <2.22e-16 |
| d-prime    | -0.9412297 | 0.15975214   | -5.891813  | 3.8198e-09 |
| referenceB | 0.5740294  | 0.14166884   | 4.051910   | 5.0801e-05 |

The standard deviation of the consumer-specific \( d' \)'s is given by:

```r
> fm3.agq$stDev * sqrt(2)
```

1.653526

The profile likelihood curve can be obtained using the profile method:

```r
> pr <- profile(fm3.agq, range = c(0.7, 1.8))
```

And then plotted using:

```r
> plpr <- plot(pr, fig = FALSE)
> plot(sqrt(2) * plpr$stDev$x, plpr$stDev$y, type = "l", xlab = expression(sigma[delta]), ylab = "Relative profile likelihood", xlim = c(1, 2.5), axes = FALSE)
> axis(1)
> axis(2, las = 1)
> abline(h = attr(plpr, "limits"))
> text(2.4, 0.17, "95% limit")
> text(2.4, 0.06, "99% limit")
```

The resulting figure is shown in Figure 2. The profile likelihood confidence intervals are obtained using:

```r
> confint(pr, level = 0.95) * sqrt(2)
```

<table>
<thead>
<tr>
<th></th>
<th>2.5 %</th>
<th>97.5 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>stDev</td>
<td>1.362331</td>
<td>2.00148</td>
</tr>
</tbody>
</table>

The probabilities that consumers prefer yoghurt A, have no preference or prefer yoghurt B can, for an average consumer be obtained by predicting from the CLMM:
The predictions for the extreme consumers have to be obtained by hand. Here we ask for
the predictions for the 5th percentile and 95th percentile of the consumer population for
reference A and B:

```r
> q95.refA <- diff(c(0, pnorm(fm3.agq$Theta - qnorm(1 - 0.05) * fm3.agq$stDev), 1))
> q05.refA <- diff(c(0, pnorm(fm3.agq$Theta - qnorm(0.05) * fm3.agq$stDev), 1))
> q95.refB <- diff(c(0, pnorm(fm3.agq$Theta - fm3.agq$beta - qnorm(1 - 0.05) * fm3.agq$stDev), 1))
> q05.refB <- diff(c(0, pnorm(fm3.agq$Theta - fm3.agq$beta - qnorm(0.05) * fm3.agq$stDev), 1))
```

Plotting follows the standard methods:

```r
> par(mar = c(0, 2, 0, 0.5) + 0.5)
> plot(1:3, pred[1:3], ylim = c(0, 1), axes = FALSE, xlab = "", ylab = "", pch = 19)
> axis(1, lwd.ticks = 0, at = c(1, 3), labels = c("", ""))
> axis(2, las = 1)
> points(1:3, pred[4:6], pch = 1)
> lines(1:3, pred[1:3])
> lines(1:3, pred[4:6], lty = 2)
> text(2, 0.6, "Average consumer")
> legend("topright", c("Reference A", "Reference B"), lty = 1:2, pch = c(19, 1), bty = "n")
> par(mar = c(0, 2, 0, 0.5) + 0.5)
> plot(1:3, q05.refA, ylim = c(0, 1), axes = FALSE, xlab = "", ylab = "",
```
Figure 3: The probabilities of preferring yoghurt A, having no preference, or preferring yoghurt B for an average consumer ($b = 0$) and for fairly extreme consumers ($b = \pm 1.64\sigma_b$).

The resulting figure is shown in Fig. 3.

7 End note

Versions details for R and the packages sensR and ordinal appear below:

```r
> sessionInfo()
```
R version 2.14.1 (2011-12-22)
Platform: x86_64-pc-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Danish_Denmark.1252 LC_CTYPE=Danish_Denmark.1252
[3] LC_MONETARY=Danish_Denmark.1252 LC_NUMERIC=C
[5] LC_TIME=Danish_Denmark.1252

attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] sensR_1.2.15 numDeriv_2010.11-1 ordinal_2012.01-19 Matrix_1.0-3
[5] lattice_0.20-0 ucminf_1.1-3 MASS_7.3-16

loaded via a namespace (and not attached):

References


Appendix E

Analysis of sensory ratings data with cumulative link models

Analysis of sensory ratings data with cumulative link models

Rune Haubo Bojesen Christensen and Per Bruun Brockhoff

Abstract: Examples of categorical rating scales include discrete preference, liking and hedonic rating scales. Data obtained on these scales are often analyzed with normal linear regression methods or with omnibus Pearson $\chi^2$ tests. In this paper we propose to use cumulative link models that allow for regression methods similar to linear models while respecting the categorical nature of the observations. We describe how cumulative link models are related to the omnibus $\chi^2$ tests and how they can lead to more powerful tests in the non-replicated setting. For replicated categorical ratings data we present a quasi-likelihood approach and a mixed effects approach both being extensions of cumulative link models. We contrast population-average and subject-specific interpretations based on these models and discuss how different approaches lead to different tests. In replicated settings, naive tests that ignore replications are often expected to be too liberal because of over-dispersion. We describe how this depends on whether the experimental design is fully randomized or blocked. For the latter situation we describe how naive tests can be stronger than over-dispersion adjusting approaches, and that mixed effects models can provide even stronger tests than naive tests. Examples will be given throughout the paper and the methodology is implemented in the authors’ free R-package ordinal.

Résumé : Exemples des ‘categorical rating scales’ comprennent la préférence discrète, ‘liking’ et ‘hedonic’. Les informations obtenaient de ces ‘rating scales’ sont souvent analysé avec les méthodes d’une ‘linear regression’ ou des tests ‘omnibus Pearson $\chi^2$’. En cet article nous proposons l’usage des modèles qu’ils s’appellent ‘cumulative link models’ qui permettent l’usage des méthodes de régression similaire aux modèles linéaires en même temps respectent la nature catégorique des observations. Nous décrivons comment les modèles ‘cumulative link models’ sont en relation avec les tests ‘omnibus Pearson $\chi^2$’ et comment ils peuvent conduire à des tests avec plus de ‘power’ en la situation ‘non-replicated’. Pour les ‘categorical ratings’ observations nous présentons une approche ‘quasi-likelihood’ et une approche ‘mixed effects’ qui sont des extensions des modèles ‘cumulative link models’. Nous comparons les interprétations de ‘population-average’ et ‘subject-specific’ qui sont basé sur ces modèles et nous discutons comment les approches différents conduisent aux tests différents. En la situation avec des réplications les tests naifs qui négligent les réplications sont souvent attendues d’être trop libérales à cause de ‘over-dispersion’. Nous nous décrivons comment cela dépend de si la conception expérimentale est randomisée complètement ou bloquée. Pour la dernière situation nous décrivons comment les tests naifs peuvent être plus fortes que les approches qui ajustent de ‘over-dispersion’, et que les modèles ‘mixed effects’ peuvent fournir les tests encore plus fortes que les tests naifs. En cet article des exemples sont présentés et la méthodologie est implémentée au R-package ordinal qui est gratuit et créé par l’auteur.

Keywords: Cumulative link models, ordinal regression models, mixed effects models, R software

Mots-clés : ,

AMS 2000 subject classifications: ,
1. Introduction

By categorical ratings data we mean data observed on an ordered categorical scale with at least two categories. This includes the common 5, 7, and 9 points preference, liking and hedonic rating scales, but excludes finite continuous scales as are used in sensory profiling. The categorical rating scales are common in sensory science as well as many other sciences where humans are used as measurement instruments [23].

There are often clear grouping structures in such data because each subject provides several observations — a concept that is known in the sensometric literature as replications. Since two observations from the same individual are likely to be more similar on average than observations from different individuals, the observations are not independent and conventional statistical tests no longer apply directly. The main objective of this paper is to propose statistical tests and models for categorical ratings data that handle grouping structures in the data appropriately. The approach we consider here is based on cumulative link models (CLMs); a well-known class of statistical models [29, 1, 2, 23].

A simple approach often described in introductory text books is to use normal linear models (regression and ANOVA) directly on the ratings under equal distance numbering of the categories. This approach can be a useful approximation if there are sufficiently many categories and not too many observations in the end categories, but it treats inherently categorical data as continuous. It is hard to quantify how this affects accuracy and consistency of parameter estimates as well as testing accuracy and power. In particular for rating scales with a small number of categories, linear models are inappropriate. A more appealing approach is to treat the observations rightfully as categorical as we do in this paper.

The conventional omnibus $\chi^2$-statistics treat data as categorical, but they do not utilize the ordering of the categories. In section 2 it will be described how cumulative link models utilize this ordering and that they often lead to stronger tests than the omnibus tests.

Tests for replicated categorical data were considered by [19], who proposed the Dirichlet-Multinomial (DM) model. Conceptually this model is equivalent to the beta-binomial model [18, 8] for multinomial rather than binomial observations. The idea is to adjust conventional statistical tests for over-dispersion. Although the DM model is applicable to ordinal data, it does not take advantage of the ordered nature of the observations.

The first approach to handling replications in categorical ratings data that we discuss is akin to the DM model in that it adjusts standard errors for over-dispersion. The amount of over-dispersion is estimated in a quasi-likelihood framework for cumulative link models. In contrast to the DM model, this approach respects the ordinal nature of the observations.

The second approach to handling replications that we propose is based on cumulative link mixed models (CLMMs) which include random effects for the grouping variable [3]. Conceptually this is an extension of linear mixed models to ordinal observations, but computationally this model class turns out to be much more complicated. Model specification and interpretation also turns out to be more complex partly due to the discrete nature of the observations and partly due to the fact that the model is nonlinear in its parameters. Due to the nonlinearity of the link function, the two approaches that we propose lead to different interpretations. The mixed models have so-called subject-specific interpretations while the over-dispersion adjusted models have population-average interpretations. The quasi-likelihood approach is a simple alternative to the more satisfying, but
also more complicated, framework of cumulative link mixed models.

Cumulative link models were also considered for analysis of data from the A-not A with sureness protocol in [13]. Replicated A-not A with sureness data are also replicated ordinal data, and while such data can also be analyzed with the methods we describe, a full treatment of the analysis of this type of data is not within the scope of this paper.

In section 2 we outline cumulative link models, we describe their relation to standard omnibus $\chi^2$ tests and the advantages of cumulative link models over these tests. We also describe a latent variable interpretation of cumulative link models that connects these with Thurstonian models. In section 3 we describe a quasi-likelihood approach to handle replicated ratings data and describe similarities and differences to the DM model. In section 4 we describe cumulative link mixed models for replicated ratings data and contrast this approach to the quasi-likelihood approach and the DM model. Most emphasis is given to the approach of cumulative link mixed models because we find that this gives the most appealing and flexible framework for modeling replicated ratings data. We end with discussions in section 5. Examples are given throughout the paper illustrating the different approaches on data from the literature. A software implementation of the methodology described in this paper is available in the \texttt{R}-package \texttt{ordinal} [11] developed by the authors freely available for the statistical software \texttt{R} [35].

2. Cumulative link models for non-replicated ratings data

In this section we outline standard cumulative link models that do not account for replications. We describe how association, e.g. product differences, can be tested in CLMs and we establish the connection to the conventional $\chi^2$-statistics. We also describe an appealing latent variable interpretation of CLMs.

2.1. Outline of cumulative link models

A cumulative link model for an ordinal variable, $Y_i$ that can fall in $J$ categories is a linear model for a transformation of cumulative probabilities, $\gamma_{ij}$ through a link function:

$$P(Y_i \leq j) = \gamma_{ij} = F(\theta_j - x_i^T \beta) \quad i = 1, \ldots, n \quad j = 1, \ldots, J$$

where the intercept parameters

$$-\infty \equiv \theta_0 \leq \theta_1 \leq \ldots \leq \theta_{J-1} \leq \theta_J \equiv \infty$$

are ordered, $F$ is the so-called inverse link function and $x_i^T$ is a $p$-vector of regression variables for the parameters, $\beta$. The linear model, $x_i^T \beta$ is assumed to apply in the same way across all response categories as it does not depend on $j$. A typical choice of link function is the probit link, $F^{-1} = \Phi^{-1}$, where $\Phi$ is the standard normal cumulative distribution function. We will adopt this choice throughout and motivate it in section 2.5. While the linear model, $x_i^T \beta$ is known as the location structure, the cumulative link model may also be extended with a scale structure, $\exp(z_i^T \xi)$ so that the resulting location-scale cumulative link model cf. [15, 2, 13] reads

$$\gamma_{ij} = F\left(\frac{\theta_j - x_i^T \beta}{\exp(z_i^T \xi)}\right) \quad i = 1, \ldots, n \quad j = 1, \ldots, J$$

(3)
The cumulative link model (1) is illustrated in Fig. 1 where $F = \Phi$ and $J = 4$ is adopted. The horizontal displacement of the three curves is determined by the values of $\theta_j$ for $j = 1, \ldots, J - 1$. The cumulative probabilities of an observation falling in each of the response categories can be read of the vertical axis for a value of the linear model, $x_i^T \beta$. The lines for $j = 0$ and $j = 4$ are horizontal straight lines at 0 and 1 by definition.

The ordinal response variable, $Y_i$ can be represented by the vector $Y_i^* = (Y_{i1}^*, \ldots, Y_{ij}^*, \ldots, Y_{iJ}^*)$ where $Y_{ij}^* = 1$ if $Y_i = j$ is observed and zero otherwise. $Y_i^*$ is said to follow the multinomial distribution $Y_i^* \sim \text{multinom}(1, \pi_i)$, where $\pi_i$ is the probability parameter vector for the $i$th observation with elements $\pi_{ij} = P(Y_i = j) = P(Y_{ij}^* = 1)$. The parameters satisfy $\sum_{j=1}^{J-1} \pi_{ij} = 1$ and are linked to the cumulative probabilities by $\gamma_{ij} = \sum_{h=1}^{j} \pi_{ih}$, or equivalently $\pi_{ij} = \gamma_{ij} - \gamma_{i,j-1}$.

The probability mass function for this multinomial distribution is the multivariate extension of the Bernoulli probability mass function $P(Y^* = y^*) = \prod_{i=1}^{n} \prod_{j=1}^{J} \pi_{ij}^{y_{ij}}$, so the log-likelihood function can be expressed as

$$\ell(\alpha; y) = \sum_{i=1}^{n} w_i \sum_{j=1}^{J} y_{ij}^* \log \pi_{ij}$$

where $w_i$ is a potential weight for the $i$th observation and $\alpha$ is a vector of all parameters.

### 2.2. Testing in cumulative link models

In this section approaches to tests of association in cumulative link models are outlined. We will consider the situation in which $k = 1, \ldots, K$, $K \geq 2$ products are rated on an ordinal scale with $j = 1, \ldots, J$, $J \geq 2$ categories with respect to preference, liking or some other aspect of interest.
The objective is to assess if and how ratings differ among products. We will assume that $k$ index rows and $j$ index columns in the resulting two-way multinomial table.

Tests of association in these two-way multinomial tables can be done via likelihood ratio tests. The likelihood ratio statistic is $LR = -2\{\ell_0(\hat{\alpha};y) - \ell_1(\hat{\alpha};y)\}$ for the comparison of two nested models $m_0$ and $m_1$ and $\hat{\alpha}$ is the ML estimates under the models. The likelihood ratio statistic asymptotically follows a $\chi^2$-distribution with degrees of freedom equal to the difference in the number of parameters for the models being compared. For binomial and multinomial observations, this statistic can also be expressed as

$$G^2 = 2 \sum_{k,j} e_{1kj} \log \frac{e_{1kj}}{e_{0kj}}$$

(4)

where $e_{1kj}$ and $e_{0kj}$ are the expected counts under models $m_1$ and $m_0$ [2, 30]. For ordinal data each row in the table is a multinomial vector which has its sum fixed by design. The expected counts are therefore given by $e_{kj} = \pi_{kj}r_k$, where $\pi_{kj}$ is the fitted probability in cell $(kj)$ and $r_k = \sum_j o_{kj}$ is the sum of the observed counts in row $k$. A closely related and often very similar statistic [2, 30] is Pearson’s statistic:

$$X^2 = \sum_{k,j} \frac{(e_{0kj} - e_{1kj})^2}{e_{0kj}}$$

(5)

These two statistics measure the discrepancy between the models $m_1$ and $m_0$ and are related through the power-divergence family [16].

Another statistic which is generally inferior to these two statistics is the Wald statistic [32, 33]. To test the existence of an effect described by a parameter vector, $\alpha$ of length $p$, the multivariate Wald statistic [46] reads

$$W = \hat{\alpha}^T \text{Cov}(\hat{\alpha})^{-1} \hat{\alpha}$$

(6)

which follows asymptotically a $\chi^2$-distribution with $p$ degrees of freedom under the null hypothesis and $\text{Cov}(\hat{\alpha})$ is the variance-covariance matrix of the parameters at their maximum likelihood estimates. For scalar $\alpha$ this can be simplified to $\sqrt{W} = \hat{\alpha}/\text{se}(\hat{\alpha})$ which follows asymptotically a standard normal distribution.

### 2.3. Connection to conventional $\chi^2$ statistics

In this section the connection between testing in cumulative link models and conventional omnibus $G^2$ and $X^2$ tests is explored. The omnibus $\chi^2$-tests can be written as in eq. (4) and eq. (5), where $e_1$ are the observed cell counts and $e_0$ are the expected cell counts given by the familiar formula $e_{0kj} = r_k \cdot c_j/N$, where $c_j = \sum_k e_{1kj}$ are the column totals and $N = \sum_{k,j} e_{1kj}$ is the overall sum. The statistics asymptotically follow a $\chi^2$-distribution on $(J-1) \cdot (K-1)$ degrees of freedom. Formally, the omnibus tests assume the following null and alternative hypotheses for our setting:

$$H_0 : \quad Y_k^* \sim \text{multinom}(m_k, \pi)$$

$$H_1 : \quad Y_k^* \sim \text{multinom}(m_k, \pi_k)$$

(7)

where $H_0$ specifies that the multinomial probability does not depend on $k$ with $J-1$ parameters, and $H_1$ specifies that the multinomial probability depends on $k$ with $(J-1) \cdot K$ parameters. The
difference in the number of parameters is \((J - 1) \cdot (K - 1)\), further, the expected counts under model \(m_1\) corresponding to \(H_1\) are exactly the observed counts, so testing the hypotheses in (7) is equivalent to application of the omnibus \(G^2\) and \(X^2\) tests.

The models implied by the hypotheses in (7) can be written as cumulative link models; \(m_0 : \gamma_j = \Phi(\theta_j)\) and \(m_1 : \gamma_{jk} = \Phi(\theta_{jk})\), where the cumulative probabilities, \(\gamma\) are linked to \(\pi\) as described in section 2.1. The model, \(m_0\) implied by \(H_0\) is known as the null model because it describes no other structure than that imposed by design, and model \(m_1\) implied by \(H_1\) is known as the full model because it completely describes the observed data with no residual degrees of freedom.

One of the main benefits of cumulative link models is that models intermediate between the null and full models can easily be specified and this often leads to stronger tests of product differences or other associations.

A cumulative link model that specifies a location difference, i.e. an additive shift on the probit scale reads

\[
\gamma_{jk} = \Phi(\theta_j - c_k) \quad j = 1, \ldots, J \quad k = 1, \ldots, K \geq 2
\]

where \(c_k\) describes the effect of the \(k\)th product. This model uses \((J - 1) + (K - 1)\) degrees of freedom, which, for \(J > 2\), is less than the full model given by \(H_1\) and therefore a model intermediate to the null and full models.

A model that specifies location as well as scale differences, i.e. additive and multiplicative effects on the probit scale reads

\[
\gamma_{jk} = \Phi\left\{ (\theta_j - c_k)/g_k \right\} \quad j = 1, \ldots, J \quad k = 1, \ldots, K \geq 2
\]

where \(g_k\) is the multiplicative effect of the \(k\)th product. This model uses \((J - 1) + 2(K - 1)\) degrees of freedom which is less than the full model if \(J > 3\) and equal to the full model if \(J = 3\). For \(J \geq 4\) a comparison of model (9) to the full model can be considered a test of differences of higher order than location and scale differences. In general the comparison of a particular working model to the full model is a goodness-of-fit (GOF) test of that model. Recall that an insignificant GOF test does not imply that the model fits well, only that the test had not enough power to provide a significant result. On the other hand a model based on plenty of data can yield a significant GOF test while still being useful and possibly an appealing model for the data generating mechanism — consequently GOF tests may be used rather informally.

Usually differences of higher order than location and scale are hard to identify and often even scale differences are negligible. The discrepancies between location and null models will therefore often be comparable in size to the omnibus \(G^2\) and \(X^2\) statistics but on fewer degrees of freedom and therefore provides a more powerful test.

An approach related to cumulative link models is that of decomposition of \(\chi^2\) statistics. The basic idea is that the omnibus statistics can be decomposed into orthogonal components each having a \(\chi^2\) distribution such that all components with appropriate degrees of freedom add up to the omnibus test. One degree of freedom tests for location and scale differences can be constructed in this way. [2, sec. 3.3.3] gives a brief description of the basic idea and [31] is a thorough description and discussion of a particular decomposition. Similar ideas are described by [36] and [37] and briefly considered in [5, sec. 5.3.2].

In contrast to the cumulative link models, the nonparametric approach does not easily generalize to the regression framework and to replicated data. While tests merely describe the degree of
TABLE 1. Ratings for a replicated paired degree-of-difference test adopted from [4]. Data are aggregated over assessors.

<table>
<thead>
<tr>
<th>pair</th>
<th>rating(^a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>concordant</td>
<td>45 40 15</td>
</tr>
<tr>
<td>discordant</td>
<td>36 34 30</td>
</tr>
</tbody>
</table>

\(^a\): 1 means identical and 3 means different

evidence of association, a model based approach also makes it possible to investigate the nature of association; the direction of differences and the strength of association, see [2, sec. 3.3.6 and 3.4] for further discussion.

2.4. Example 1

In this example we compare various \(\chi^2\) tests. [4] describes a replicated paired degree-of-difference test where 25 subjects each assess four concordant and four discordant product pairs. The subjects were asked to rate the degree of difference between the sample product pairs on a three point rating scale, where 1 means identical and 3 means different. In this example we will ignore the grouping structure in the data and analyze the data as if they were independent. The data are summarized in Table 1.

A test of differences in ratings between concordant and discordant sample pairs is a test of product differences. Using eq. (5), we find that the omnibus Pearson \(\chi^2\)-test statistic is \(X^2 = 6.49\). On 2 degrees of freedom, we may find using tables or statistical software that the \(p\)-value is \(p = 0.039\). Similarly, by application of eq. (4) the omnibus likelihood ratio \(\chi^2\)-test statistic is \(G^2 = 6.59\), which on \(df = 2\) gives \(p = 0.037\). Since the full model is equivalent to the location-scale model (9), the same test could be obtained as a likelihood ratio test of the comparison of model (9) and the null model.

The likelihood ratio test of a location difference is obtained by comparing the null model with model (8). This leads to \(G^2 = 4.70\), \(df = 1\), \(p = 0.030\) and therefore a slightly stronger test than the omnibus tests. The likelihood ratio test of scale and higher order differences while controlling for location differences is \(G^2 = 1.88\), \(df = 1\), \(p = 0.170\). This test can be obtained as the likelihood ratio test of models (8) and (9) or, since the \(\chi^2\) statistics are additive, as the difference between the omnibus \(G^2\) test and the likelihood ratio test of a location difference: \(6.59 - 4.70 = 1.88\) save for rounding errors. Observe also that the likelihood ratio test of scale and higher order differences can be considered a GOF test of the location model (8).

The main discrepancy in these data is due to location differences, while there is no evidence of differences in scale and higher order moments. The test of location differences is a stronger test than the omnibus tests because the main discrepancy in the table can be summarized as a location difference on only one degree of freedom. This is a fairly typical result that is often even more pronounced in situations with more response categories.

2.5. Latent variable interpretation

The cumulative link model can be interpreted as a model for a continuous latent variable. Suppose for instance that preference for a particular product type, \(S\) can be described by a normal linear
model: $S \sim N(\mu, \sigma^2)$, where $\mu$ describes structural differences in preference, for instance the average difference in preference between consumers from different regions and $\sigma$ is the residual standard deviation. The variation in preference could be due to differences in product samples, differences in perception of the samples or variations in preference. Preference, $S$ is not observed directly — only a categorized version, $Y$ is observed. This latent variable interpretation is conceptually similar to the Thurstonian model of paired preferences [42, 43, 44]. Suppose that $Y$ is observed in the $j$th category if $S$ falls between the thresholds $\theta_j$ and $\theta_j$ obeying (2), then the cumulative probabilities can be expressed as a function of the model parameters: $\gamma_j = \Phi(\theta_j - \mu_i)$. This is the cumulative link model with a probit link, where $\mu_i$ can be described by a general linear predictor: $\mu_i = x_i^T \beta$ as in eq. (1). In this model $\mu_i$ refers to a location difference relative to the origin, $\mu^0$ and scale, $\sigma$ of the latent distribution; $\mu_i = (\mu_i^* - \mu^0)/\sigma$; similar arguments appear in [10] and [13].

The latent variable interpretation of a cumulative link model is illustrated in Fig. 2 where a probit link and $J = 4$ is adopted. The three thresholds, $\theta_1, \ldots, \theta_3$ divide the area under the curve into four parts each of which represent the probability of a response falling in the four response categories. The thresholds are fixed on the scale, but the location of the latent distribution, and therefore also the four areas under the curve, change with $x_i^T \beta$. Assuming other latent distributions lead to other link functions, for example, assuming that $S$ has a logistic distribution leads to a logit link. The location-scale model (cf. eq. (3) and (9)) arise if the spread of the latent distribution is also allowed to depend on $i$.

Likelihood ratio tests of effects are often fairly unaffected by the choice of link function and often very large amounts of data are necessary to distinguish between the links in terms of goodness-of-fit [22]. Different link functions, however, lead to different parameter estimates and interpretations can differ.
2.6. Example 2

The test of location differences in example 1, section 2.4, is a test of $H_0 : c_2 - c_1 = 0$ versus $H_1 : c_2 - c_1 \neq 0$ in model (8) with $K = 2$ and $J = 3$. This implies the latent distributions; $S_k \sim N(\mu_0 + c_k, \sigma)$, where the absolute location, $\mu_0$ and scale $\sigma$ are unknown and not estimable from data, but the maximum likelihood estimate of the location difference $c_2 - c_1$ is 0.3462 with standard error 0.160. The maximum likelihood estimates of the threshold parameters are $\hat{\theta} = (-0.073, 0.939)$.

3. Adjusting for over-dispersion in replicated ratings data

It was recognized by [19] that standard statistical tests are not appropriate when grouping structures violate the assumption of independent observations. They proposed to adjust the test statistics by an amount related to the degree of over-dispersion in the data relative to what would be expected for independent observations. The degree of over-dispersion is estimated in a Dirichlet-multinomial (DM) model where the multinomial probabilities are allowed to vary. A disadvantage of this approach for rating data is that it treats ordinal data as unordered. We present in this section an alternative approach of adjusting tests for over-dispersion within models that take advantage of the ordered nature of the ratings.

3.1. Quasi-likelihood approach

A well-known way of modeling over-dispersed discrete data is the quasi-likelihood approach [47, 30]. The basic idea is to model the population mean of the observations with a linear predictor through a link function. The amount of over-dispersion is estimated by comparing the observed variation with the variation that would be expected if the observations were independent. This approach conceptually amounts to estimating parameters with a standard cumulative link model, but adjusting the variance-covariance matrix of the parameter estimates by multiplication with an over-dispersion parameter $\phi$.

Over-dispersed cumulative link models are specified in terms of the first two moments of the distribution of the observations and so does not assume a full likelihood specification including higher order moments. This means that likelihood ratio tests are unavailable, but Wald tests of individual parameters and multivariate Wald tests of model terms can be constructed. Approximate $F$-tests for model terms can also be constructed that are similar to likelihood ratio tests for likelihood based models, but these $F$-tests tend to be rather conservative, so we do not consider them further. See [14] for construction of these $F$-tests in binomial models; see also [30] and [45] for relevant discussion. In this approach, observations are assumed to follow a quasi-multinomial distribution with expectation $E[Y] = m\pi$ and covariance $\text{Cov}[Y] = \phi m(\text{diag}(\pi) - \pi\pi^T)$, where the over-dispersion parameter, $\phi$ distinguishes the distribution from a genuine multinomial distribution which has $\phi \equiv 1$. A cumulative link model is assumed to describe the mean structure in the observations and the variance-covariance matrix of the parameters is given by $\text{Cov}[\alpha] = \phi G_{\alpha}$, where $G_{\alpha}$ is the variance-covariance matrix of $\alpha$ in a standard full likelihood approach. The quasi-likelihood approach therefore amounts to inflating the standard errors of the parameter estimates with $\sqrt{\phi}$. 

Soumis au Journal de la Société Française de Statistique
File: manus.tex, compiled with jsfds, version : 2009/12/09
date: September 28, 2012
TABLE 2. Replicated degree of liking data from 104 subjects reported by [19] aggregated over subjects.

<table>
<thead>
<tr>
<th>city</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>New York</td>
<td>9 26 15 120 154</td>
</tr>
<tr>
<td>San Francisco</td>
<td>5 21 28 129 117</td>
</tr>
</tbody>
</table>

a: 1 means “dislike very much” and 5 means “like very much”.

There are two standard approaches to estimate $\phi$ related to the $G^2$ and $X^2$ statistics [33, 30]:

$$\hat{\phi}_G = \frac{2}{n-p} \sum_{k,j} e_{1kj} \log \frac{e_{1kj}}{e_{0kj}}$$

$$\hat{\phi}_P = \frac{1}{n-p} \sum_{k,j} \left( \frac{e_{0kj} - e_{1kj}}{e_{0kj}} \right)^2$$

where $n = (J-1) \cdot K$ is the total number of degrees of freedom, $p$ is the number of parameters in the model, $e_{1kj}$ are the observed cell counts and $e_{0kj}$ are the expected cell counts under the model. This corresponds to a generalized estimation equation (GEE) approach assuming a so-called independence working correlation model [20, sec. 3.5]. The estimators are only valid when the multinomial table is not sparse; as a general rule the expected frequencies should be at least five. There are generally only minor differences between the two $\phi$-estimators. A considerable difference is an indication that the model is inappropriate and tests in the model should not be trusted. When this occurs the expected frequencies are small or important structures have been ignored in the data.

Under the quasi-likelihood model a modified Wald statistic, $W^* = W/\hat{\phi}$ is used instead of the standard Wald statistic (6) to test association.

3.2. Example 3

In [19] the degree degree of liking among consumers in a replicated rating experiment conducted in New York and San Francisco was considered. A five-point liking scale was adopted where 1 means “dislike very much” and 5 means “like very much”. 54 subjects from New York and 50 subjects from San Francisco were included in the study and each of the subjects evaluated six samples of the product. The main objective is to consider whether there is a difference in liking between cities. Data are summarized in Table 2 aggregated over subjects.

The omnibus Pearson and likelihood ratio tests applied directly to Table 2 yield $X^2 = 10.07$, df = 4, $p = 0.039$ and $G^2 = 10.15$, df = 4, $p = 0.038$ indicating a difference between the cities with respect to liking.

The joint test of location and scale differences while ignoring the grouping structure (replications) in the location-scale model (9) with $K = 2$ and $J = 5$ is $LR = 6.71$, df = 2, $p = 0.035$. The multivariate Wald test for the same hypothesis yields $W = 7.31$, df = 2, $p = 0.023$. The likelihood ratio test for higher order differences is $LR = 3.44$, df = 2, $p = 0.179$, so there is no evidence of more than location and scale differences.

In [19] a Dirichlet-Multinomial (DM) model was fitted to the data from each of the two cities and obtain estimates of over-dispersion correction parameters, which employed in a (bivariate)
Wald test yields \( p \)-value = 0.38 of the difference in liking between the two cities. They conclude that when adjusting for over-dispersion, there is no evidence of a difference in liking between the two cities.

The estimate of \( \phi_G \) based on model (9) is the likelihood ratio statistic for the test of higher order differences scaled by the residual degrees of freedom, \( n - p = 4 - 2 = 2 \), i.e. \( \phi_G = 3.44/2 = 1.72 \). The Wald statistic for the joint test of location and scale differences is \( W^* = 7.31/1.72 = 4.25 \), which on 2 degrees of freedom gives \( p = 0.119 \). This test is adjusted for the over-dispersion caused by the replications and is therefore more appropriate than the naive test; consequently the naive test assuming independent observations is too liberal. The tests of regional differences based on quasi-likelihood and the DM model lead to the same conclusion of no evidence of a difference.

4. Cumulative link mixed models for replicated ratings data

In this section we consider an extension of cumulative link models where random effects are included in the location part of the predictor. As such it can also be viewed as an extension of linear mixed models to ordered categorical observations. This framework is more flexible than the quasi-likelihood approach and allows for a more insightful modeling of grouping structures. Cumulative link mixed models is a member of a class of models sometimes referred to as multivariate generalized nonlinear mixed models [20]. The latent variable interpretation carries over to the mixed versions of cumulative link models and if the probit link is assumed, the model amounts to a standard linear mixed model for the latent variable. A cumulative link mixed model with a single random effect term can be expressed as

\[
\gamma_{ijl} = F(\theta_j - \mathbf{x}_{il}' \mathbf{\beta} - b_i) \\
i = 1, \ldots, n \quad l = 1, \ldots, l_i \quad j = 1, \ldots, J
\]

where it is assumed that the conditional distribution of the observations given the realizations of the random effects is multinomial and the random effects are normally distributed

\[
(Y_{il} | B_i = b_i) \sim \text{Multinom}(1, \pi_{il}) \quad B_i \sim N(0, \sigma_b^2)
\]

The \( \mathbf{\beta} \) and \( \mathbf{\theta} \) parameters describe the structure in the conditional distribution of the observation, and \( \sigma_b \) describes the heterogeneity in the population. This model is akin to a normal linear mixed model where the response is treated as ordinal rather than continuous and normally distributed. If the inverse link function, \( F \) is taken to be the standard normal cumulative distribution function, this model corresponds to assuming the following linear mixed model for the latent scale:

\[
S_{il} = \mathbf{x}_{il}' \mathbf{\beta} + b_i + e_{il} \quad E_{il} \sim N(0, \sigma^2) \quad B_i \sim N(0, \sigma_b^2)
\]

This is possibly the simplest model for the latent scale that accounts for the grouping structure in the data.

The population spread, \( \sigma_b \) is estimated relative to the spread of the latent scale, so it can be interpreted as a ratio of between-to-within subject variation. Observe also that the size of \( \sigma_b \) changes with the link function. This is not only because this means another mapping from the linear predictor to the probability scale, but primarily because the variance of the residuals (cf. eq. (10)) change with the distributional assumptions entailed by the link function. For instance a
logit link corresponds to assuming a logistic distribution for the latent scale, and since the standard logistic distribution has variance $\pi^2/3$, the estimated $\sigma_b$ will be scaled by approximately $\pi/\sqrt{3}$ compared to the estimate obtained with a probit link.

The log-likelihood function for the models we consider may be written as

$$
\ell(\theta, \beta, \sigma_b; y) = \sum_{i=1}^{n} \log \int_{\mathbb{R}} p(y_i|b_i) p(b_i) \, db_i
$$

where $p(y_i|b_i)$ is the conditional multinomial probability mass function of the observations given the random effects, and $p(b_i)$ is the (marginal) normal density of the random effects. The log-likelihood is a sum of $n$ independent contributions since observations from different individuals are assumed independent.

Estimation of CLMMs is complicated by the fact that the integral in eq. (11) does not have a closed form solution. Several different approximations have been proposed and two of the most popular are the Laplace approximation and adaptive Gauss-Hermite quadrature (AGQ) [27, 34, 39, 26]. The Laplace approximation is a fast and reasonably accurate approximation while AGQ is computationally more intensive, but it has the advantage that the accuracy can be increased by adding more quadrature nodes. Often the Laplace approximation is sufficiently accurate while essentially exact estimates can often be obtained from AGQ with a few, e.g. 5–10 nodes. Following [26] we recommend that the Laplace approximation is used initially; the final model may be estimated accurately with AGQ by increasing the number of nodes until the parameter estimates do not change by any relevant amount.

The Laplace approximation and AGQ are implemented in R-package ordinal [11] for CLMMs and AGQ is also implemented in the NLMIXED procedure for SAS [24].

### 4.1. Attenuation effects

A mixed effects model is known as a *conditional model* because the model is formulated for the conditional distribution of the response given the random effects. This means that the parameters of the model apply at the level of subjects and not at the population level, so these parameters are known as *subject-specific* parameters.

Models like the quasi-CLM and DM model are known as marginal models since these models are formulated for the marginal distribution of the response $E_B[Y]$. Usually in such models the correlation structure is treated as a nuisance and only needed to obtain inference for the mean structure. Since the marginal distribution is modeled directly, the parameters of these models apply at the population level and are denoted *population-average* parameters [17, 2, 21].

Marginal models do not model individuals explicitly like conditional models, where a subject-distribution is assumed. Conditional models are models for the data-generating mechanism whereas quasi models are not full distributional descriptions.

While it is not possible to obtain subject-specific interpretations from a marginal model, it is possible to obtain population-average interpretations from a conditional model because a particular conditional model implies a marginal model [49, 48]. Marginal predictions and population-average parameters can therefore be obtained in two general ways: 1) by modeling the marginal distribution directly or 2) by obtaining the marginal predictions and parameters implied by a conditional
model. Often these two population-average parameter sets are of similar magnitude and usually lead to the same inference. Consequently, conditional cumulative link mixed models constitute a richer framework than the marginal models.

In normal linear mixed models subject-specific and population-average parameters coincide, but in cumulative link mixed models, and generalized linear mixed models in general, the population-average parameters implied by a conditional model are *attenuated*, i.e. smaller in absolute size, relative to the subject-specific parameters. In the CLMM with a probit link, the expectation over the random effects distribution, i.e. the implied marginal model, can be derived explicitly, for details see appendix A:

\[
E_B[\gamma_{jl}] = E_B[\Phi(\theta_j - x_{jl}\beta - b_j)] \\
= \Phi(\theta_{pa} - x_{jl}\beta_{pa})
\]

where \(\theta_{pa} = \theta / \sqrt{1 + \sigma_b^2}\) and \(\beta_{pa} = \beta / \sqrt{1 + \sigma_b^2}\) are the population-average parameters implied by the conditional model.

The attenuation effect is illustrated in Fig. 3. Each of the dashed curves represent the latent distribution, \(f(S_i)\) for an individual at two predictor values, \(x_1\) and \(x_2\). At \(x_2\) the latent distributions are shifted an amount, \(\beta\) relative to \(x_2\). The solid curves are the latent distributions at the population level which are averaged over individuals. Due to the variation among individuals, the population-

**FIGURE 3. Illustration of the attenuation effect of a CLMM in terms of latent distributions.**
average distributions have higher variance than the subject-specific distributions. The relative shift of the curves, that is the size of the shift relative to the spread of the curves, is therefore smaller for the population-average distributions than for the subject-specific distributions.

If all individuals assess a single sample, the indexes \( i \) and \( l \) coincide (cf. eq. (10) and (12)) and the variance components, \( \sigma_i^2 \) and \( \sigma_l^2 \) are completely confounded. Thus, if there is heterogeneity among individuals, the estimate of \( \beta \) from the non-replicated design (or if individual heterogeneity is not accounted for in a replicated design), \( \beta^{na} \) is attenuated, i.e. too small in absolute size. While the standard tests are valid for non-replicated designs even if there is variation among individuals, the parameter estimates are not consistent and too small in absolute size.

### 4.2. Tests in marginal and conditional models

In marginal models, inter-individual variation will always translate into over-dispersion, inflation of standard errors and therefore more conservative tests of e.g. product differences. In conditional models this is not always the case. Not only can the naive test; the test ignoring replications all together, be more appropriate than the test in a marginal model with inflated standard errors, the test in a conditional model can also be even more powerful than the naive test. In some cases the naive test will even be unreasonable and a more appropriate test is provided by the conditional model.

This may happen in randomized block settings, that is, in situations with crossed factors, as in example 1, where each consumer evaluated both concordant and discordant product pairs. The randomized block setting is the most common structure for consumer preference studies — although typically with only one evaluation for each combination of product and consumer — or even less in incomplete settings. In the example here, there are replications on top of the randomized blocks, but this is not of key importance for the point to be made here. As opposed to this we have the purely nested (“completely randomized”) situation illustrated in example 3, with a grouping of the consumers as the effect of interest — again with additional replications within consumers on top of this.

In the randomized block settings, the proper test for product/treatment differences does not include the block (main) effect — it is removed from the error — this is the main idea of making a blocked experiment. In normal linear models with complete data for an unreplicated randomized block experiment, the sums of squares (SS) decompose into: \( SS(\text{total})=SS(\text{block})+SS(\text{treat})+SS(\text{error}) \), and the treatment effect is tested against mean square for error. Ignoring blocks in this setting would lead to an error term based on \( SS(\text{block})+SS(\text{error}) \) rather than \( SS(\text{error}) \) which in turn leads to a conservative test. Only in situations with either a very weak block effect or a very high number of products relative to the number of blocks, this is not a major problem, but indeed for the typical consumer experiment the pooling of the block effect into the error will grossly affect the analysis.

In analyses of binomial and ordinal data the tests cannot be expressed exactly in terms of mean squares like this, but only approximately so. Consider for example that for binomial data with observations away from the extremes and high enough binomial denominator, a normal approximation analysis would give more or less the same results as a standard logistic regression analysis. Clearly, even though the data come in a binomial or ordinal form, it will generally be inadequate to pool the (random) block effect into error, which is exactly what a marginal analysis...
corresponds to. To summarize, the simple over dispersion approach entailed by marginal models is not well suited to handle random effect models other than purely nested ones. This point is not commonly realized nor even discussed in the literature. A clear advantage of mixed effects models is that they lead to correct tests irrespective of experimental design and effect sizes.

4.3. Example 4

In this example we will revisit the paired degree-of-difference test from example 1 in section 2.4. A Stuart-Maxwell test adjusted for over-dispersion was suggested in [4]. This test gave $\tilde{X}_p^2 = 3.85$, df = 2, $p = 0.146$, which is more conservative than the tests that assumed independent observations. Similarly, for a Wald test in a quasi-CLM we find $\hat{\phi}_Q = 1.88$, so $W^* = 1.58$, df = 1, $p = 0.114$ with essentially the same conclusion. A cumulative link mixed model that allows for subject-specific effects reads

$$
\gamma_{ijk} = \Phi(\theta_j - p_k - b_i) \quad j = 1, \ldots, 5 \quad k = 1, 2 \quad i = 1, \ldots, 100
$$

(13)

Observe that product and subject factors are crossed in the sense of section 4.2, so we can expect the test of product differences to be as strong, or perhaps stronger, in the mixed effects model in comparison with the naive test. The likelihood ratio test of $p_k$ in model (13) is $LR = 5.84$, df = 1, $p = 0.016$, which provides strong evidence of a product difference. Not only is the test stronger than the adjusted Stuart-Maxwell test and the Wald test from a quasi-CLM, it is also stronger than the naive tests reported in example 1 for the same data where individual differences were ignored.

4.4. Tests of random effects terms in cumulative link mixed models

Likelihood ratio tests can be used to test fixed-effects model terms in the same way for cumulative link mixed models as in cumulative link models — tests of random effect terms is a bit more complicated. A likelihood ratio test of a random-effects term is a test of the following hypotheses for the variance parameter:

$$
H_0 : \sigma_b = 0 \quad \text{versus} \quad H_1 : \sigma_b > 0 .
$$

(14)

Observe that the test is one-sided, since the random effects standard deviation is non-negative. The usual asymptotic theory for the likelihood ratio statistic, $LR$ dictates that the $LR$ asymptotically follows a $\chi^2_1$-distribution with one degree of freedom. However, since the $\sigma_b$ is on the boundary of the parameter space, the usual asymptotic theory does not hold. Following [38, 40] the $LR$ more closely follows an equal mixture of $\chi^2$-distributions with zero degrees of freedom (a point mass distributions) and one degree of freedom. The $p$-value from this test can be obtained by halving the $p$-value from the test assuming $LR \sim \chi^2_1$. This adjusted test can be motivated by the following: for a single parameter, we can consider the likelihood root statistic, $r = \text{sign}(\hat{\sigma}_b - \sigma_0) \sqrt{LR}$; the signed square root of the likelihood ratio statistic [7, 33], which under the usual asymptotic theory follows a standard normal distribution. Here, $\sigma_0$ is the value of $\sigma_b$ under the null hypothesis and $\hat{\sigma}_b$ is the maximum likelihood estimate of $\sigma_b$. The $p$-value from the one-sided test of the hypotheses in (14) can be computed as $p = 1 - \Phi(r)$ and is exactly the $p$-value from the adjusted likelihood ratio test.
Wald tests of the variance parameter can also be constructed, but since the profile log-likelihood function is only approximately quadratic when \( \hat{\sigma}_b \) is not small and well defined, such tests cannot be recommended [32, 6]. Confidence intervals for parameters should preferably be constructed from profile likelihood functions rather than from inverted Wald tests as is for instance implemented in the R-package \texttt{ordinal} [11].

### 4.5. Example 5

In this example we continue the analysis of the paired degree-of-difference test from example 1 in section 2.4 and illustrate how inference for the assessor population can be conducted.

The estimated location difference between the two products is 0.404 with standard error 0.168 and the random-effects standard deviation is \( \hat{\sigma}_b = 0.614 \). Observe that estimate and standard error of the location parameter are larger as expected. The thresholds estimates are \( \hat{\theta} = (-0.073, 0.939) \). The relative profile likelihood for \( \sigma_b \) in Fig. 4 displays the evidence in the data about this parameter. The 99% confidence interval includes zero while the 95% confidence interval does not. While a random-effects spread of zero has some support, it is not likely to be considerably larger than one. The one-sided hypotheses in (14) yields \( p = 0.014 \), but the significance of \( \sigma_b \) was already visible from the profile likelihood in Fig. 4.

The variance parameter can be interpreted as the variation in the subjects’ use of the response scale, i.e. the variation in the thresholds among the subjects. Roughly 95% of the population will be within \( \pm 2\hat{\sigma}_b = 1.23 \), so the shift of the thresholds will span roughly 1.23 units among the population. In comparison, the distance between the thresholds is 1.01 and the product effect is 0.404, so the variation in the population is considerable compared to the distance between the thresholds and the product effect.
4.6. Example 6

In this section we continue the analysis of the consumer liking example in section 3.2 where we found very weak evidence of a difference in liking ($p = 0.119$) in a Wald test adjusting for over-dispersion. A cumulative probit mixed model for these data reads

$$\gamma_{jk} = \Phi\left(\frac{\theta_j - c_k - b_{(i(k))}}{g_k}\right) \quad i = 1, \ldots, n_k \quad j = 1, \ldots, J \quad k = 1, 2,$$

(15)

where the parenthesized index on $b$ indicates that subjects are nested within cities. The joint $LR$ test of $c_k$ and $g_k$ in model (15) gives $p = 0.35$ which is close to the result by [19], confirming that there is no evidence of a difference in liking between the two cities. In this case subjects are nested in cities and the naive test is liberal compared to the tests that take account of replications in line with the discussion in section 4.2. Further, tests from the conditional and marginal models lead to equivalent conclusions. There is, however, a considerable variation among consumers in their perception of the liking scale. The maximum likelihood estimate of $\sigma_b$ is 0.944 in model (15). The normalized profile likelihood in Fig. 5 confirms that the spread is well-determined. The likelihood root statistic from the one-sided test of $\sigma_b$ is 12.84 corresponding to a $p$-value of essentially zero (around $5 \times 10^{-38}$).

The cumulative link model (CLM), the cumulative link model with over-dispersion adjusted standard errors (quasi-CLM) and the cumulative link mixed model (CLMM) are summarized in Table 3. The parameter estimates for the CLM and quasi-CLM are identical and only the standard errors differ reflecting the adjustment for over-dispersion in the quasi-model. The estimated location and threshold parameters are larger in absolute measures for the CLMM in line with the discussion in section 4.1. Also observe that the standard error of the location parameter is
TABLE 3. Maximum likelihood parameter estimates (standard errors) in models for the city preference data.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Model (9) CLM</th>
<th>Model (9) quasi-CLM</th>
<th>Model (15) CLMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_2$</td>
<td>$-0.183(0.084)$</td>
<td>$-0.183(0.111)$</td>
<td>$-0.231(0.208)$</td>
</tr>
<tr>
<td>$g_2$</td>
<td>$-0.185(0.088)$</td>
<td>$-0.185(0.115)$</td>
<td>$-0.154(0.098)$</td>
</tr>
<tr>
<td>$\sigma_b$</td>
<td></td>
<td></td>
<td>0.944</td>
</tr>
<tr>
<td>$\theta$</td>
<td>$-1.93, -1.28, -0.98, 0.056$</td>
<td>$-1.93, -1.28, -0.98, 0.056$</td>
<td>$-3.00, -1.86, -1.34, 0.14$</td>
</tr>
<tr>
<td>log-lik.</td>
<td>$-714.15$</td>
<td></td>
<td>$-658.73$</td>
</tr>
</tbody>
</table>

TABLE 4. The probabilities of rating a product in the five categories for the city preference data.

<table>
<thead>
<tr>
<th>Segment</th>
<th>dislike very much</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>like very much</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>0.022</td>
<td>0.075</td>
<td>0.069</td>
<td>0.399</td>
<td>0.434</td>
</tr>
<tr>
<td>Population-average$^a$</td>
<td>0.015</td>
<td>0.079</td>
<td>0.083</td>
<td>0.398</td>
<td>0.424</td>
</tr>
<tr>
<td>Subject-specific</td>
<td>0.001</td>
<td>0.029</td>
<td>0.063</td>
<td>0.515</td>
<td>0.392</td>
</tr>
<tr>
<td>5% percentile subject</td>
<td>0.077</td>
<td>0.344</td>
<td>0.219</td>
<td>0.335</td>
<td>0.025</td>
</tr>
<tr>
<td>95% percentile subject</td>
<td>&lt;0.001</td>
<td>&lt;0.001</td>
<td>0.001</td>
<td>0.078</td>
<td>0.920</td>
</tr>
</tbody>
</table>

$^a$: Population-average predictions from CLM and CLMM models coincide.

larger in the CLMM than in the CLM. In the quasi-model the standard errors are inflated by the same amount while only the standard error of the location parameter is appreciably bigger in the CLMM.

The parameter estimates for the CLMs have population-average interpretations, i.e. they correspond to the effects that we see at the population level. The parameter estimates in the CLMM have subject-specific interpretations. This is not particularly important for the location and scale differences in this example since these effects are small and insignificant, but it makes a difference in the interpretation of the fitted probabilities. For simplicity of exposition we ignore location and scale differences and consider a CLMM only accounting for subject differences, $\gamma_j = \Phi(\theta_j - b_i)$. The threshold estimates are $(-3.11, -1.88, -1.32, 0.27)$, and the random effects spread estimate is $\hat{\sigma}_b = 1.02$. The probabilities that ratings fall in each of the five categories are presented in Table 4. The first line presents the raw sample proportions. The fitted probabilities from a CLM with no predictors, $\gamma_j = \Phi(\theta_j)$, were identical to the population-average predictions from the CLMM only accounting for subject differences to three digits. These are presented in the second line and are seen to correspond very well to the raw sample proportions. The third line are the fitted probabilities for an average subject, i.e. with $b_i = 0$, which is distinctly different from the probabilities at the population level. From the sample or population estimates we might be tempted to conclude that an average individual would have the highest probability of responding in the “like very much” category because the highest probability is associated with this category, but this is not correct. This kind of subject-specific interpretation should be based on the conditional model predictions presented in line three of Table 4. From this line we see that an average subject is most likely to respond in the fourth category and not the fifth. To illustrate the variation between subjects, the ratings of the 5% and 95% percentiles in the distribution of subjects has been included as well reflecting how subjects that are relatively extreme would tend to rate samples. People that like the products the least primarily use the middle categories, while virtually no one would primarily use the “dislike very much” category. On the other hand people that like the products the most almost exclusively rate the products in the “like very much” category.
5. Discussion

In this paper we have shown how cumulative link models can be used for sensory tests on categorical ratings data. We have described how cumulative link models relates to standard omnibus \( \chi^2 \) tests and how cumulative link models often lead to stronger tests of association because the ordinal nature of ratings data can be utilized. We have suggested two extensions of cumulative link models for replicated data and compared these approaches to the Dirichlet-Multinomial model suggested in [19]. Our first suggestion is a quasi-cumulative link model which leads to Wald tests adjusted for over-dispersion, and our second suggestion is a cumulative link mixed model that explicitly models the population of subjects. When the factor of interest is crossed with the subject factor, marginal models in adjusting for over-dispersion can lead to tests that are weaker and more conservative than naive tests while more correct tests like those of (conditional) mixed models are actually stronger than naive tests. So while approaches adjusting for over-dispersion are not always appropriate, mixed (conditional) models lead to appropriately sized tests irrespective of experimental design. The mixed model can also provide insight into how subjects use the rating scale and can provide subject-specific as well as population-average interpretations. All models discussed in this paper can be fitted with the authors’ freely available R-package \texttt{ordinal} [11].

It was shown in [9] how several common discrimination protocols (m-AFC, duo-trio, triangle and A-not A) can be identified as generalized linear models. This makes it possible to adjust analyses for the effects of e.g. gender differences or varying concentrations of an additive. In this way sensory discrimination and preference protocols are combined with statistical models that enhance the models with a general regression framework. In the same line of work it was shown in [13] how the identification of the Thurstonian model for the A-not A with sureness protocol as a cumulative link model with a probit link could allow the analysis of such data to take account of explanatory variables describing the assessors/consumers or the experimental conditions. In this paper we have shown how cumulative link mixed models accommodate replications via random effects. Cumulative link mixed models also extend naturally with a general regression framework and makes it possible to model and control for the effect of explanatory variables — these extensions are also supported by the \texttt{ordinal} package [11].

In more complicated settings, e.g. in larger consumer preference studies including for instance many consumers, many products and possibly many sessions, it may be of interest to include two or more cross-classified factors as random terms in the model. This is a computational challenge for some of the current software available in the simpler normal linear mixed model and currently not possible in more complicated models such as the cumulative link mixed model for ordinal data.

One of the examples considered a degree-of-difference rating experiment. This protocol is an extension of the same-different protocol to a rating scale, and while a Thurstonian model has been derived for the same-different protocol, see [28] and [12] for further details, we are not aware of derivations of the Thurstonian model for the degree-of-difference protocol, see, however [25] for a discussion. The cumulative link model is not a Thurstonian model for degree-of-difference ratings data per say.
Appendix A: Marginal parameters in a CLMM with a probit link

Following [41] taking the expectation with respect to the distribution of $B$ gives

\[
E_B[\gamma_{ij}] = E_B[\Phi(\theta_j - x_{ij}\beta - b_i)] \\
= E_B[P(Z \leq \theta_j - x_{ij}\beta - b_i)] \\
= P \left( Z \leq \frac{\theta_j - x_{ij}\beta}{\sqrt{1 + \sigma^2_B}} \right) \\
= \Phi \left( \frac{\theta_j - x_{ij}\beta}{\sqrt{1 + \sigma^2_B}} \right) \\
= \Phi(\theta^m_j - x_{ij}\beta^m)
\]

References


Replicated Categorical Ratings Data


Appendix F

Paired preference data with a no-preference option — statistical tests for comparison with placebo data.

Paired preference data with a no-preference option —
statistical tests for comparison with placebo data

Rune Haubo Bojesen Christensen\textsuperscript{a,*}, John M Ennis\textsuperscript{b}, Daniel M Ennis\textsuperscript{b}, Per Bruun Brockhoff\textsuperscript{a}

\textit{a DTU Informatics, Statistics section, Technical University of Denmark, Richard Petersens Plads, Building 305, DK-2800 Kongens Lyngby, Denmark.}

\textit{b The Institute for Perception, Richmond, VA, USA}

Abstract

It is well-established that when respondents are presented with identical samples in a preference test with a no preference option, a sizable proportion of respondents will report a preference. In a recent paper (Ennis & Ennis, 2012, Accounting for no difference/preference responses or ties in choice experiments. \textit{Food Quality and Preference, 23}, 13-17) noted that this proportion can depend on the product category, have proposed that the expected proportion of preference responses within a given category be called an identicality norm, and have argued that knowledge of such norms is valuable for more complete interpretation of 2-Alternative Choice (2-AC) data. For instance, these norms can be used to indicate consumer segmentation even with non-replicated data. In this paper, we show that the statistical test suggested by Ennis & Ennis (2012) behaves poorly and has too high a type I error rate if the identicality norm is not estimated from a very large sample size. We then compare five $\chi^2$ tests of paired preference data with a no preference option in terms of type I error and power in a series of scenarios. In particular, we identify two tests that are well behaved for sample sizes typical of recent research and have high statistical power. One of these tests has the advantage that it can be decomposed for more insightful analyses in a fashion similar to that of ANOVA F-tests. The benefits are important because they enable more informed business decisions, particularly when ingredient changes are considered for cost-reduction or health initiative purposes.

*Corresponding author

\textit{Email address: rhbc@imm.dtu.dk (Rune Haubo Bojesen Christensen)}

Preprint submitted to Food Quality and Preference September 28, 2012
1. Introduction

Difference testing is as relevant as ever given the numerous ingredient change projects currently underway for cost-reduction or health-initiative purposes. Thus, there is presently much interest in interpreting difference testing results in as reliable, as meaningful, and as powerful a manner as possible (Bi, Lee & O’Mahony, 2011; Brockhoff & Christensen, 2010; Christensen & Brockhoff, 2009; Christensen, Cleaver & Brockhoff, 2011; Christensen, Lee & Brockhoff, 2012; Ennis & Jesionka, 2011; Ennis & Ennis, 2012b; Hautus, Shepard & Peng, 2011; Ishii, Kawaguchi, O’mahony & Rousseau, 2007; Lee, van Hout & Hautus, 2007; van Hout, Hautus & Lee, 2011).

In a recent paper, Ennis & Ennis (2012a) developed the idea of an identicality norm for 2 Alternative Choice (2-AC) data. The 2-AC protocol is a 2-AFC protocol with a no difference option and is technically identical to the paired preference test with a no preference option. The identicality norm is obtained by conducting a paired preference test with a no preference option with identical products — the idea is similar to that of using a placebo drug in a medical trial. The expected distribution over “Prefer A”, “No Preference” and “Prefer B” for the identical products then constitutes the identicality norm.

The identicality norm can be useful in a situation where a paired preference test with a no preference option has been conducted, but the products appear to be approximately equally preferred and a conventional statistical test, e.g. a Pearson test does not show significant differences in preference between the two products in question. However, the products might appear equally preferred if the consumer sample consists of two segments with opposite preferences; in this case preferences may approximately balance out in the sample and the products appear to be equally preferred while in fact most consumers have a preference for either of the products.

Ennis & Ennis (2012a) observed that the data table can be compared to an identicality norm in a Pearson $\chi^2$ test, and that this test can indicate if opposing segments of preference balance out over the sample as just described.

---

1For the remainder of this article, we will refer only to no preference votes, for simplicity.
But, as we will see in this paper, the statistic proposed by Ennis & Ennis (2012a) only performs well if the identicality norm is based on a placebo experiment with a very large sample size. Otherwise, as we will show, the failure of this statistic to take into account the variability within the placebo experiment leads to an unacceptably large type I error.

The purpose of this paper is to advance the statistical analysis of 2-AC data with placebo experiments by comparison of five statistical tests. To this end we propose two tests that are well behaved for sample sizes typical of recent research (Alfaro-Rodriguez, Angulo & O’Mahony, 2007; Chapman & Lawless, 2005; Kim, Lee, O’Mahony & Kim, 2008; Marchisano, Lim, Cho, Suh, Jeon, Kim & O’Mahony, 2003) and have high statistical power. One test has the feature that asymptotically, as the sample size for the placebo experiment approaches infinity, the $\chi^2$ test suggested by Ennis & Ennis (2012a) is obtained. The other test has the advantage that it may be conveniently decomposed into directional and tie effects in a fashion similar to that of ANOVA $F$-tests.

In section 2 example data from Ennis & Ennis (2012a) are re-analyzed illustrating that the uncertainty in the placebo experiment is not taken into account. In section 3 five test statistics are presented, problems with the genuine Pearson test are exposed and alternative tests are suggested. In section 4 these five tests are compared in terms of type I error rate and power in a series of scenarios. In section 5 we end with discussion and recommendations. All computations were done in R (R Development Core Team, 2011) and the code to perform all simulations are available in the online supplements [Please insert appropriate link here].

2. $\chi^2$ tests with identicality norms

To motivate the adjustments we suggest to the $\chi^2$ test that was originally suggested by Ennis & Ennis (2012a), we will use the example presented in section 4 in Ennis & Ennis (2012a). In this example it is assumed that the following triplet of data have been obtained (25, 15, 60 for “Prefer A”, “No Preference”, “Prefer B”), and that the identicality norm can be assumed to be 40%, 20% and 40% for those three response options.

Ennis & Ennis (2012a) in essence suggest that we compute expected values as $100 \cdot (0.4, 0.2, 0.4) = (40; 20, 40)$ and compare these to the observed
Table 1: Observed counts for example in section 2.

<table>
<thead>
<tr>
<th></th>
<th>Prefer A</th>
<th>No Preference</th>
<th>Prefer B</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preference experiment</td>
<td>25</td>
<td>15</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>Placebo experiment</td>
<td>40</td>
<td>20</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>65</td>
<td>35</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 2: Expected values for the observed counts in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Prefer A</th>
<th>No Preference</th>
<th>Prefer B</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preference experiment</td>
<td>32.5</td>
<td>17.5</td>
<td>50.0</td>
<td>100</td>
</tr>
<tr>
<td>Placebo experiment</td>
<td>32.5</td>
<td>17.5</td>
<td>50.0</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>65.0</td>
<td>35.0</td>
<td>100.0</td>
<td>200</td>
</tr>
</tbody>
</table>

values, (25, 15, 60) in a Pearson $\chi^2$ test. The test statistic is

$$X^2 = \frac{(25 - 40)^2}{40} + \frac{(15 - 20)^2}{20} + \frac{(60 - 40)^2}{40}$$

$$= 5.625 + 1.250 + 10.00 = 16.875.$$ 

Comparing this value to a $\chi^2$ distribution (with 2 degrees of freedom) yields a $p$-value of 0.00022 as also found by Ennis & Ennis (2012a).

Observe that this test does not depend on the sample size involved in setting the identicality norm, hence the identicality norm is inherently assumed to be known without error. If the identicality norm is determined without any uncertainty all is well, but if an experiment with identical products as described in the introduction was used to obtain the identicality norm, it will be determined with some uncertainty, and it is desirable to take account of that uncertainty in the statistical test.

Intuitively we expect that if the identicality norm is obtained using a large sample size, it is accurately determined and the results should not change much. If, on the other hand, the identicality norm is obtained from a small sample size, the norm is more uncertain and it should be harder to get a significant result.

Now assume, for instance, that the identicality norm in our example was determined from a placebo experiment with 100 observations. We can then arrange the data in the $2 \times 3$ table shown in Table 1. The corresponding table of expected values are given in Table 2. As an example, the expected value in the (1, 1) cell is obtained as $100 \cdot \frac{65}{200} = 32.5$ since the sum in the first row is 100, the sum in the first column is 65 and the total sum is 200.
Table 3: Pearson $\chi^2$ statistic and p-value for a range of sample sizes for the placebo experiment.

<table>
<thead>
<tr>
<th>n</th>
<th>Statistic</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.55</td>
<td>0.46118</td>
</tr>
<tr>
<td>20</td>
<td>2.80</td>
<td>0.24619</td>
</tr>
<tr>
<td>30</td>
<td>3.85</td>
<td>0.14601</td>
</tr>
<tr>
<td>40</td>
<td>4.74</td>
<td>0.09371</td>
</tr>
<tr>
<td>50</td>
<td>5.50</td>
<td>0.06393</td>
</tr>
<tr>
<td>60</td>
<td>6.17</td>
<td>0.04578</td>
</tr>
<tr>
<td>70</td>
<td>6.76</td>
<td>0.03410</td>
</tr>
<tr>
<td>80</td>
<td>7.28</td>
<td>0.02624</td>
</tr>
<tr>
<td>90</td>
<td>7.75</td>
<td>0.02074</td>
</tr>
<tr>
<td>100</td>
<td>8.18</td>
<td>0.01677</td>
</tr>
<tr>
<td>1000</td>
<td>15.15</td>
<td>0.00051</td>
</tr>
<tr>
<td>$10^4$</td>
<td>16.68</td>
<td>0.00024</td>
</tr>
<tr>
<td>$10^5$</td>
<td>16.86</td>
<td>0.00022</td>
</tr>
<tr>
<td>$10^9$</td>
<td>16.87</td>
<td>0.00022</td>
</tr>
</tbody>
</table>

Computing the Pearson $\chi^2$ test on these tables now yields

$$X^2 = \frac{(25 - 32.5)^2}{32.5} + \frac{(40 - 32.5)^2}{32.5} + \ldots + \frac{(40 - 50.0)^2}{50.0}$$

$$= 8.18$$

The number of degrees of freedom are $(2 - 1) \cdot (3 - 1) = 2$, and in comparison with the $\chi^2$ distribution we obtain a p-value of 0.0168. The Pearson $\chi^2$ test for association in this table is a test for whether the paired preference test data are in compliance with the placebo data, therefore essentially the same hypotheses are tested as in the test suggested by Ennis & Ennis (2012a). This p-value is larger reflecting that the uncertainty in the identicality norm is taken into account. While the sample size for the identicality norm is still large enough that the test is significant, this will change if a smaller sample size for the placebo experiment is assumed.

Table 3 shows the value of the Pearson $\chi^2$ statistic and p-value for a range of sample sizes used to determine the identicality norm. The table shows that the smaller the sample size used for setting the identicality norm, the larger the p-value. Had, for instance, the sample size for the identicality norm been 50, the test would not have been significant on the 5% limit. Table 3 also
Table 4: Example data for section 3.

<table>
<thead>
<tr>
<th></th>
<th>Prefer A</th>
<th>No Preference</th>
<th>Prefer B</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placebo experiment</td>
<td>83</td>
<td>42</td>
<td>75</td>
<td>200</td>
</tr>
<tr>
<td>Preference experiment</td>
<td>36</td>
<td>14</td>
<td>50</td>
<td>100</td>
</tr>
</tbody>
</table>

shows that the test suggested by Ennis & Ennis (2012a) is obtained in the limit as the sample size used to determine the identicality norm approaches infinity. For smaller sample sizes, however, the evidence that the paired preference data do not comply with the identicality norm is weaker reflecting the larger uncertainty in the determination of the identicality norm.

3. Presentation of test statistics

In paired preference data with a no preference option we would typically test whether preference for one product is higher than for the other product: we denote this type of effect a directional effect. If data from a placebo experiment are available, we can also assess if there is a difference in the probability of a tie in the preference experiment and the placebo experiment. We will denote such an effect a tie effect. In the following we will consider statistical tests for three kinds of effect structures in the data:

- A a tie effect and no directional effect,
- B a directional effect with no tie effects and
- C the joint effects of tie effects and directional effects.

Different statistical tests will be sensitive to these structures to a different extent and we will consider each situation separately in the following. In this section we will consider a data set for examples summarized in Table 4.

All the tests we will consider are based on the Pearson $\chi^2$ statistic:

$$X^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

for different tabulations of the data and therefore with different values for the observed ($O_i$) and expected ($E_i$) values. Here $i$ runs through all the entries of the appropriate table.

Following the notation in Ennis & Ennis (2012a), we denote the probability that an individual responds “No Preference” by $s$. If a subject expresses
a preference, a second parameter $p$ denotes the probability that the subject prefers the first product (A) over the second product (B). Collectively we then for each subject have that

\[
\begin{align*}
P(\text{"Prefer A"}) &= p(1 - s) \\
P(\text{"No Preference"}) &= s \\
P(\text{"Prefer B"}) &= (1 - p)(1 - s)
\end{align*}
\]

In the following we let $s_0$ and $p_0$ denote the parameters of the placebo experiment and $s_1$ and $p_1$ denote the parameters of the preference experiment.

3.1. The $\chi^2$ test of directional effects

The directional test is the only test that does not use an identicality norm or data from a placebo experiment. The counts observed in “Prefer A” and “Prefer B” are simply considered a $1 \times 2$ contingency table and a conventional Pearson test applied. Note that this is identical to Putter’s test (Ennis & Ennis, 2012a and Christensen et al., 2012).

The hypothesis involved in the directional test is whether preference for A and B products are the same, that is, whether preference for one over another is one half:

\[
\begin{align*}
H_0 : & \quad p_1 = 0.5 \\
H_A : & \quad p_1 \neq 0.5
\end{align*}
\]

3.1.1. Example 1: test for directional effects

In the test for directional effects we use only the data from the preference experiment. The expected counts (under the null hypothesis of no directional effect) are $E_1 = E_2 = (36 + 50)/2 = 43$. The Pearson $\chi^2$ statistic for the test of directional effects is therefore:

\[
X^2_{\text{direc}} = \frac{(36 - 43)^2}{43} + \frac{(50 - 43)^2}{43} = 2.28
\]

Compared with a $\chi^2_1$-distribution on one degree of freedom this gives a $p$-value of 0.131.

3.2. A test for difference in the probability of ties

Like the test for directional effects, the test for tie effects is a one degree of freedom test. This test is targeted at differences in the probability of a
ti in the placebo and preference experiments. Data are arranged in a $2 \times 2$ contingency table with data from placebo and preference experiments in each row, the sum of the counts for “Prefer A” and “Prefer B” in the first column, and the “No Preference” counts in the second column. A conventional Pearson $\chi^2$ test is performed on the table.

The hypothesis involved in the test for ties can be described as a test for the equality of the probability of “No Preference” in placebo and preference experiments, that is, a test for the equality of $s_0$ and $s_1$:

$$H_0 : \ s_0 = s_1$$

$$H_A : \ s_0 \neq s_1.$$  

3.2.1. Example 2: test for tie effects

To compute the test statistic for tie effects, we have to reorganize the data as follows: The counts for “Prefer A” and “Prefer B” are summed up into a single “Preference” category to give the tabulation in Table 5. The corresponding expected counts are given in Table 6. The $\chi^2$ test statistics is computed as

$$\chi^2_{tie} = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

$$= \frac{(158 - 162.67)^2}{162.67} + \frac{(86 - 81.33)^2}{81.33} + \ldots + \frac{(14 - 18.67)^2}{18.67}$$

$$= 2.15,$$

and compared with a $\chi^2$-distribution on one degree of freedom this gives $p = 0.142.$
3.3. The genuine Pearson $\chi^2$ test

We now consider the usual or genuine Pearson test. This test is the first of three statistics for tests of joint effects, that is, considering as well directional effects as tie effects. The genuine Pearson $\chi^2$ test is applied directly to the $2 \times 3$ table of counts as illustrated in section 2. This test accounts for the uncertainty in the placebo experiment.

The genuine Pearson test can be described as the joint test that preference for A over B is the same in placebo and preference experiments, and that the probability of a “No Preference” is the same in placebo and preference experiments, that is, that $s_0 = s_1$ and $p_0 = p_1$:

$$H_0 : \ s_0 = s_1 \text{ and } p_0 = p_1$$

$$H_A : \ s_0 \neq s_1 \text{ and } p_0 \neq p_1.$$ 

However, this is not the set of hypotheses we are usually interested in assessing. The reason is that $p_0$ is allowed to vary freely and we know for a fact that $p_0 = 0.5$. The genuine Pearson statistic is in this sense the wrong statistic for tests involving identicality norms. The modified Pearson test and the pooled test that we describe in the following are two suggestions for dealing with this problem.

3.3.1. Example 3: the genuine Pearson test

In the genuine Pearson test we would compute the expected frequencies directly from the observed data and compute the $\chi^2$ statistic in the usual way:

$$X_{\text{genuine}}^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

$$= \frac{(83 - 79.33)^2}{79.33} + \frac{(36 - 39.67)^2}{39.67} + \ldots + \frac{(50 - 41.67)^2}{41.67}$$

$$= 4.76.$$ 

Compared with a $\chi^2_2$-distribution on two degrees of freedom this gives $p = 0.0926$.

3.4. A modified Pearson $\chi^2$ test

Observe that in the placebo experiment we obtained different preference counts for the A and B products by chance, as shown in Table 4. However,
since we know that the true probabilities for “Prefer A” and “Prefer B” are identical, we can estimate the identicality norm and use these derived counts or “pseudo observations” in the $\chi^2$ test. This would address the $p_0 \neq 0.5$ problem with the genuine Pearson test exposed in the previous section. Operationally, we simply average the counts for “Prefer A” and “Prefer B” in the table of observed counts.

Essentially, the modified Pearson test assumes that preference for A over B to be one half, that is, $p_0 = 0.5$. The hypotheses involved in the modified Pearson test can therefore be described as

$$H_0 : s_0 = s_1 \text{ and } p_1 = 0.5$$
$$H_A : s_0 \neq s_1 \text{ and } p_1 \neq 0.5$$

which means that the probability of preference for A over B in the placebo experiment, $p_0$ is not involved in the hypotheses and we can assume that $p_0 = 0.5$ under both null and alternative hypotheses. In contrast to the hypotheses for the genuine Pearson test, the hypotheses for the modified Pearson test explicitly tests whether preference for A over B is one half in the preference experiment.

To compute the test statistic for the modified Pearson test, we need the expected counts, and these are implied by the data structure under the null hypothesis. We therefore need a table in which the probability of “No Preference” is the same for both experiments and in which preference for A over B is one half in each experiment. To obtain this, we can use the expected “No Preference” counts from the test for ties. We then distribute the counts where a preference was expressed equally to the “Prefer A” and “Prefer B” categories separately for the placebo and preference data.

A special feature of the modified Pearson test is that asymptotically, that is, as the sample size in the placebo experiment increases, the test suggested by Ennis & Ennis (2012a) is obtained.

### 3.4.1. Example 4: the modified Pearson test

The modified Pearson $\chi^2$ test constructs pseudo data by averaging the preference counts from the placebo experiment. Pseudo observations for our example are given in Table 7.

The expected counts are shown in Table 8 in which the “No Preference” counts are obtained from the expected counts for the tie test in Table 6. The expected counts for “Prefer A” and “Prefer B” are half the expected counts
Table 7: Pseudo observations for the modified Pearson test in example 4.

<table>
<thead>
<tr>
<th></th>
<th>&quot;Prefer A&quot;</th>
<th>&quot;No Preference&quot;</th>
<th>&quot;Prefer B&quot;</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placebo</td>
<td>79</td>
<td>42</td>
<td>79</td>
<td>200</td>
</tr>
<tr>
<td>Preference</td>
<td>36</td>
<td>14</td>
<td>50</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 8: Expected counts for the modified Pearson test in example 4.

<table>
<thead>
<tr>
<th></th>
<th>&quot;Prefer A&quot;</th>
<th>&quot;No Preference&quot;</th>
<th>&quot;Prefer B&quot;</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placebo</td>
<td>81.33</td>
<td>37.33</td>
<td>81.33</td>
<td>200</td>
</tr>
<tr>
<td>Preference</td>
<td>40.67</td>
<td>18.67</td>
<td>40.67</td>
<td>100</td>
</tr>
</tbody>
</table>

for the “Preference” counts in the tie test also in Table 6. The $\chi^2$ statistic is then computed from Tables 8 and 9 in the usual way:

$$X_{\text{modif}}^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

$$= \frac{(79 - 81.33)^2}{81.33} + \frac{(36 - 40.67)^2}{40.67} + \ldots + \frac{(50 - 40.67)^2}{40.67}$$

$$= 4.56,$$

and compared with a $\chi^2$-distribution on two degrees of freedom this gives $p = 0.102$.

3.5. Relation between genuine and modified Pearson tests

Intuitively, the modified Pearson $\chi^2$ test seems more appropriate than the genuine Pearson $\chi^2$ test, since we know that with identical products, the probability of preference for the products should be the same. In this way relevant information is included. This exposes a concern with the genuine Pearson test: if by chance we obtain a much larger proportion of preferences for one of the products in the placebo experiment, the genuine Pearson test will tend to be significant, while the modified test will not. Thus, the modified test should be able to guard against some falsely significant test results. To illustrate the problem with the genuine Pearson test, consider for example the situation where the data in Table 9 are observed.

Here the observed frequency of “Prefer B” is considerably higher than the observed frequency of “Prefer A” in the placebo data by chance — we know that the true probabilities for “Prefer A” and “Prefer B” are the same since the products are identical. For these data the genuine Pearson $\chi^2$
Table 9: Example data for comparison of genuine and modified Pearson tests.

<table>
<thead>
<tr>
<th>Test</th>
<th>Prefer A</th>
<th>No Preference</th>
<th>Prefer B</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placebo</td>
<td>52</td>
<td>56</td>
<td>92</td>
<td>200</td>
</tr>
<tr>
<td>Preference</td>
<td>72</td>
<td>58</td>
<td>70</td>
<td>200</td>
</tr>
</tbody>
</table>

The test yields $X^2_{\text{genuine}} = 6.25$ with a significant $p$-value of 0.044, while neither the directional test, nor the test for ties are close to significance ($X^2_{\text{direc}} = 0.0282$ with $p = 0.867$ and $X^2_{\text{tie}} = 0.0491$ with $p = 0.825$). In comparison the modified Pearson test yields $X^2_{\text{modif}} = 0.077$ with $p$-value 0.962 so it successfully guards against this false significance.

The example shown here is obviously somewhat extreme for the purpose of illustration, but note that this type of “false significances” will actually occur with a frequency of 1.5%\(^2\). Observe also that this problem is avoided in the test for difference in the probability of ties presented above; since the counts for “Prefer A” and “Prefer B” are added up, any differences between the two occurring by chance are irrelevant in this test.

3.6. A pooled test

We now propose a pooled $\chi^2$ test statistic for a joint test of a difference in the probability of ties and directional effects. The proposed test is based on the observation that the test for ties and the test for directional effects described above are independent $\chi^2$ tests each on one degree of freedom\(^3\). We may therefore form a joint test statistic by summation of the test statistics for the test for ties and the test for directional effects. This new test statistic should then be compared to a $\chi^2_2$ distribution with two degrees of freedom. Just like the modified Pearson test outlined above, the pooled test avoids the problem with the genuine Pearson test described in section 3.5 above.

The hypotheses involved in the new test can be described as a test for the equality of the probability of “No Preference” in placebo and preference experiments.
Table 10: ANOVA-like analysis of the data in Table 1 in which the Pooled test statistic is split up into contributions from tie effects and directional effects.

<table>
<thead>
<tr>
<th>Test</th>
<th>$\chi^2$</th>
<th>df</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pooled test</td>
<td>15.28</td>
<td>2</td>
<td>0.000481</td>
</tr>
<tr>
<td>Tie effects</td>
<td>0.87</td>
<td>1</td>
<td>0.352</td>
</tr>
<tr>
<td>Directional effects</td>
<td>14.41</td>
<td>1</td>
<td>0.000147</td>
</tr>
</tbody>
</table>

experiments, i.e. $s_0 = s_1$ jointly with a test that preference for A over B is the same in the preference experiment, $p_1 = 0.5$. Observe that these tests are free of considerations of the preference probability, $p_0$ in the placebo experiment. The null and alternative hypotheses are therefore the same as that for the modified Pearson test:

$$H_0 : s_0 = s_1 \text{ and } p_1 = 0.5$$
$$H_A : s_0 \neq s_1 \text{ and } p_1 \neq 0.5.$$

In contrast to the modified Pearson test, the pooled test does not equal the test suggested by Ennis & Ennis (2012a) asymptotically as the sample size in the placebo experiment increases. On the other hand, the pooled test has the advantage that it decomposes into two independent $\chi^2$ tests for directional and tie effects, respectively.

3.6.1. Example 5: the pooled $\chi^2$ test

For our example data we compute the pooled test statistic as

$$X^2_{\text{pooled}} = X^2_{\text{direc}} + X^2_{\text{tie}} = 4.43$$

which compared to a $\chi^2_2$-distribution on two degrees of freedom gives $p = 0.109$.

3.6.2. Example 6: An ANOVA-like analysis

A clear benefit of the pooled test is that it decomposes into independent contributions from tie effects and directional effects. This is similar to ANOVA analyses in which $F$-tests for several degree of freedom terms can be split into multiple one degree of freedom tests. To illustrate this, we will use the example data from section 4 in (Ennis & Ennis, 2012a) that we also used in section 2 and are shown in Table 1.

The ANOVA-like presentation in Table 10 shows how the $\chi^2$ statistic for the pooled test is decomposed into $\chi^2$ statistics for the directional effects.
and tie effects. The pooled test is highly significant showing that clearly there is some structure in the preference data different from the the placebo data. The decomposition further reveals that this is almost entirely due to directional effects ($\chi^2 = 14.41$) while the contribution due to tie effects is very small ($\chi^2 = 0.87$).

Recall that we analyzed these data in section 2 with the genuine Pearson test. Now, observe that the pooled test in Table 10 reveals much stronger evidence than did the genuine Pearson test ($\chi^2 = 8.18$, $p = 0.0168$). This is a prime example of a better statistical test providing stronger evidence than a less appropriate test.

4. Comparison of $\chi^2$ tests with simulations

In this section the type I error and power of the $\chi^2$ tests discussed in section 3 are examined using simulations.

4.1. Type I errors

For assessment of type I errors the parameters are as follows: For all but the directional test, the sample size for the preference experiment is fixed at 100 and the sample size for the placebo experiment varied from 10 to 1,000,000. Similarly for the directional test, the sample size for the preference experiment is varied from 10 to 1,000,000. The probability of an answer in each of the “Prefer A”, “No Preference” and “Prefer B” categories are 35%, 30% and 35% respectively. These are the probabilities for both the placebo and preference experiments — they are equal since we are simulating under the null hypothesis.

The curves in Figure 1 illustrate the type I error as a function of increasing sample size. It is shown that the type I error of the test suggested by Ennis & Ennis (2012a) is much higher than the nominal 5% for small sample sizes but eventually reaches the 5% level for large sample sizes. The remaining test statistics all have type I error rates fluctuating closely around the nominal 5% level.

4.2. Power curves

In this section power curves are developed for six scenarios being combinations of two sample sizes for the placebo experiment and three types of effect structures in the preference experiment as shown in Table 11. The small sample size (100) for the placebo experiment is typical of recent research
while the other setting reflects a large sample size for which the identicality norm can be estimated with very little uncertainty. At this sample size the genuine Pearson test statistic is very similar to the test statistic suggested by Ennis & Ennis (2012a), cf. Table 3. Throughout we assume a sample size for the preference experiment of 100.

In scenario 1A and 2A the difference in the probability of a tie is increased from 0 to 0.5. In the notation of section 3, \( s \) is increased from 0.2 under the null hypothesis up to 0.7, cf. the horizontal axis of the top panel in Figures 2 and 3. The directional probability is constant \( p = 1/2 \). In scenario 1B and 2B the probability of a tie is constant \( s = 1/2 \) throughout, while \( p \) is decreased from 1/2 as indicated in the middle panel of Figure 2 and 3. In scenario 1C and 2C both tie effects and directional effects are present, so \( s \) is increased from 0.2 to 0.5 while at the same time the directional effect \( P(\text{"Prefer B"}) - P(\text{"Prefer A"}) \) is increased from 0 to 0.3.

Power curves are generated for the five statistics presented in section 3. The test by Ennis & Ennis (2012a) is not separately included; for small sample sizes the type I error of this test is unacceptably large (cf. Figure 1), while in the large sample size setting this test is indistinguishable from the genuine Pearson test (cf. section 2) thus it is implicitly included via the genuine Pearson test.

Finally we note that 100,000 simulations were used to compute type I error and power at each of the points in Figures 1, 2 and 3. This means that the standard error of an estimated type I error of 0.05 or a power of 95% is \( \text{se}(95\% \text{ power}) = \sqrt{0.95 \cdot 0.05/10^5} \approx 0.0007 \) thus providing plenty of precision in the type I error and curves power curves.

4.3. Comparison of power curves

The general picture from Figures 2 and 3 is that in the A scenarios where tie effects, but no directional effects are present, the test for ties is has the highest power. Similarly in the B scenarios where directional effects, but no tie effects are present, the directional test is best. In the C scenarios

<table>
<thead>
<tr>
<th>Placebo sample size</th>
<th>Structures in preference data</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Placebo sample size tie effects directional effects joint effects</td>
</tr>
<tr>
<td>1.000.000</td>
<td>1A 1B 1C</td>
</tr>
<tr>
<td></td>
<td>2A 2B 2C</td>
</tr>
</tbody>
</table>
where both tie and directional effects are present, the modified Pearson and Pooled tests generally have the highest power. Furthermore, we note that the power of the modified Pearson and Pooled tests are almost indistinguishable at all the settings we considered. Moreover, the genuine Pearson test is indistinguishable from the modified Pearson and Pooled tests in the large sample setting.

Now considering the scenarios individually, it is seen from scenarios 1A in Figure 2 and 2A in Figure 3 that the genuine Pearson, the modified Pearson and the Pooled tests are equally sensitive to tie effects with only a little power gain from using the test for ties. The test for directional effects has no power at all beyond the type I error rate.

In scenario 1B in Figure 2 the modified Pearson and Pooled tests are seen to be more sensitive to directional effects and therefore more powerful than the genuine Pearson test. However, this difference vanishes in the large sample setting, cf. scenario 2B in Figure 3.

Scenario 1C in Figure 3 is the most interesting and most relevant setting all together. Here, the modified Pearson and Pooled tests have the highest power and the remaining tests have less power.

In scenario 2C in Figure 3 the test for ties has very similar power to that of the joint tests. This is due to the relative small role that the directional test plays in this setting given the small sample size for preference experiment ($n = 100$) relative to the placebo experiment ($n = 1.000.000$).

5. Discussion and recommendations

In considering the joint tests, the modified Pearson and Pooled tests have very similar power curves and it is impossible to recommend one over the other on that basis. However, these two tests are both superior to the genuine Pearson test, as it may give false significances as discussed in section 3.5. Moreover, the genuine Pearson test has substantially less power than the modified Pearson test and the Pooled test in scenarios 1B and 1C, cf. Figure 2. The Pooled test has the advantage over the modified Pearson test that it decomposes into a contribution from tie effects and a contribution from directional effects as illustrated in example 6. This means that it is directly possible to see which of the directional and tie effects are important and which might be negligible. On this ground we find the Pooled test more insightful and recommend that it is used for comparison of 2-AC preference data with placebo data.
Appendix A. Online supplementary material

R code to do the examples and simulations in the paper.

References


Figure 1: A: Type I error for a series of statistics as a function of sample size for the identicality norm. B: Type I error for the directional test as a function of sample size for the preference test.
### Scenario 1A - Tie effects

<table>
<thead>
<tr>
<th>Power</th>
<th>Pr(A)</th>
<th>Pr(tie)</th>
<th>Pr(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>0.2</td>
<td>0.35</td>
<td>0.3</td>
<td>0.45</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>0.6</td>
<td>0.25</td>
<td>0.2</td>
<td>0.35</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td>1.0</td>
<td>0.15</td>
<td>0.2</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Test for ties: 
- Pooled test
- Modified Pearson
- Genuine Pearson
- Directional test

### Scenario 1B - Directional effects

<table>
<thead>
<tr>
<th>Power</th>
<th>Pr(A)</th>
<th>Pr(tie)</th>
<th>Pr(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>0.2</td>
<td>0.35</td>
<td>0.3</td>
<td>0.45</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>0.6</td>
<td>0.25</td>
<td>0.2</td>
<td>0.35</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td>1.0</td>
<td>0.15</td>
<td>0.2</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Test for ties: 
- Pooled test
- Modified Pearson
- Genuine Pearson
- Directional test

### Scenario 1C - Joint effects

<table>
<thead>
<tr>
<th>Power</th>
<th>Pr(A)</th>
<th>Pr(tie)</th>
<th>Pr(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>0.2</td>
<td>0.35</td>
<td>0.3</td>
<td>0.45</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>0.6</td>
<td>0.25</td>
<td>0.2</td>
<td>0.35</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td>1.0</td>
<td>0.15</td>
<td>0.2</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Test for ties: 
- Pooled test
- Modified Pearson
- Genuine Pearson
- Directional test

Figure 2: Power curves for scenario 1 where the sample size for the placebo experiment is 100
Figure 3: Power curves scenario 2 where the sample size for the placebo experiment is 1,000,000
Appendix G

Flexible estimation of nonlinear mixed models via the multivariate Laplace approximation

Flexible Estimation of Nonlinear Mixed Models via the Multivariate Laplace Approximation

Stig Bousgaard Mortensen$^{1,2}$ &
Rune Haubo Bojesen Christensen$^1$

January 18, 2010

$^1$Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Lyngby, Denmark. $^2$E-mail: sbm@imm.dtu.dk

Abstract

The multivariate Laplace approximation to the marginal likelihood is a fast and accurate approach for estimation in nonlinear mixed models. Our aim is to show how this approximation is easily implemented on a case-by-case basis in general programming environments such as R, S-plus or Matlab. The approach is very flexible compared to what is possible in standard statistical software allowing estimation of models with e.g. crossed random effects and arbitrary correlation structures for the residuals. Such models are not easily, if at all possible, fit with standard statistical software or software specially designed for nonlinear mixed models. The approach also allows graphical assessment of successful convergence and can produce profile likelihoods for selected parameters, neither of which is generally possible with standard statistical software.

Keywords: Crossed random effects, orange tree data, R.
1 Introduction

This paper is concerned with estimation of nonlinear mixed models (NLMMs) where the conditional distribution of the response given the random effects as well as the distribution of the random effects are Gaussian. The model can be expressed generally as

\[
(Y | B = b) \sim N(f(\beta, b), \Sigma(\lambda)), \quad B \sim N(0, \Psi(\psi)),
\]

where \( \beta \) are fixed regression parameters, \( b \) is a \( q \)-vector of random effects, \( \lambda \) and \( \psi \) are variance parameters parameterizing the covariance matrices \( \Sigma \) and \( \Psi \) and \( f \) is the model function. NLMMs can be viewed as a generalization of the ordinary fixed effects nonlinear model (NLM) (Bates and Watts, 1988) to include random effects, and it can be viewed as a generalization of the linear mixed model (LMM) (Laird and Ware, 1982) to allow the conditional mean to be a nonlinear function of the regression parameters. Despite these conceptually small changes, maximum likelihood estimation in NLMMs has been a fair challenge and still is to some extent. The likelihood function of a NLMM is the marginal density of the response where the random effects are integrated out

\[
L(\theta; y) = \int_{\mathbb{R}^q} p(y | \theta, b)p(b) \, db,
\]

where \( p \) denotes a normal probability density function and \( \theta = [\beta^T, \lambda^T, \psi^T]^T \) denotes the vector of fixed parameters. It is this \( q \)-dimensional integral that is difficult to solve in general, because approximations have to be invoked. The likelihood can be reduced to a multiple of integrals of lower order when the random parameters arise from only one level of grouping (indexed by \( i \), say); the model can be written as

\[
(Y_i | B_i = b_i) \sim N(f_i(\beta, b_i), \Sigma_i), \quad B_i \sim N(0, \Psi), \quad i = 1, \ldots, M,
\]

and the likelihood simplifies to a multiple of \( q_i \)-dimensional integrals

\[
L(\theta; y) = \prod_{i=1}^{M} \int_{\mathbb{R}^{q_i}} p(y_i | \theta, b_i)p(b_i) \, db_i,
\]
where \( q_i \) is the number of random effects for the \( i \)th group. In particular, the likelihood reduces to a multiple of one-dimensional integrals when only a single random effect occurs for each group in the data. We will refer to a set of random effects corresponding to a single grouping variable as a random component.

In this paper we shall be concerned with the multivariate Laplace approximation to solve the full \( q \)-dimensional integral (2) and thereby allow for any structure of the random effects: scalar or vector-valued random effects, nested, crossed or partially crossed; linear, as well as nonlinear.

Pinheiro and Bates (1995) and Vonesh (1996) studied the Laplace approximation (Barndorff-Nielsen and Cox, 1989; Tierney and Kadane, 1986) for models with a single level of grouping, and Pinheiro and Bates (2000) also studied it for models with nested random effects. Statistical software packages that fit NLMMs (e.g. \texttt{nlme} in R and S-plus (Pinheiro et al., 2008), SAS NL MIXED (SAS Institute Inc., 2004) and NONMEM (Beal and Sheiner, 2004), hereafter denoted “standard software”) are designed for models with a single level of grouping or with nested random effects. We show in this paper how the Laplace approximation can be implemented on a case-by-case basis in around 20 lines of code providing fast convergence to accurate maximum likelihood estimates (MLEs) for the general NLMM.

Convergence of NLMMs can be hard to achieve and software can be fooled to declare convergence at a local optimum rather than the global optimum or simply far from optimum due to correlation among the parameters. It is therefore important that the user is able to assess, preferably by graphical methods, that a global optimum has been reached at convergence and whether several local optima with high likelihood exist. We suggest to use pseudo-likelihood curves to facilitate this assessment because they are simple and fast to compute.

While the standard error is a convenient summary of uncertainty in a parameter estimate, it is not always appropriate for regression parameters in NLMMs due to nonlinearities (Bates and Watts, 1988) and almost never appropriate for the variance parameters. Profile likelihood curves and corresponding confidence intervals are natural alternatives to standard errors when these are inappropriate. Regrettably, we do not find them easy to obtain with standard software and it seems that the user is left with likelihood ratio tests of the parameters as the only appropriate inferential tool. Profile likelihoods and corresponding confidence
intervals for the variance parameters are easy to obtain from the estimation scheme that we propose.

To illustrate our approach to estimation of NLMMs, we use the orange tree data set that has been used repeatedly in the literature to illustrate estimation of NLMMs. These data and appropriate models for them will be presented in section 1.1. In Section 2 we outline the Laplace approximation and motivate it as a natural approximation to the marginal likelihood for NLMMs. In Section 3 we describe how estimation of NLMMs via the multivariate Laplace approximation can be achieved in few lines of code, and we illustrate the flexibility of the approach. In Section 4 we discuss profile likelihoods and assessment of convergence, and we illustrate how the accuracy of the Laplace approximation can be assessed post hoc. In Section 5 we compare our approach with standard software, and we end with a discussion and conclusions in Section 6.

We illustrate estimation using the statistical programming environment R (R Development Core Team, 2008) and include R-code in the text for illustration, but the approach can be implemented in any functional programming environment (e.g. Matlab) that provides access to a general optimizer (preferably of quasi-Newton type), finite difference approximations to Jacobians and Hessians of user defined functions and allows basic matrix operations.

The complete R-code to produce all fits and figures can be downloaded from http://imm.dtu.dk/~sbm/nlmm/.

1.1 The orange tree data set and appropriate models

To illustrate our approach to estimation of NLMMs, we use a study of the growth of orange trees reported by Draper and Smith (1981, p. 524), see the appendix, where the circumference of five trees is measured at seven time points. This data set has been used by Lindstrom and Bates (1990) to illustrate their algorithm, by Pinheiro and Bates (1995) in their comparison of estimation methods and in Wolfinger (1999) to illustrate the NLMIXED procedure. A logistic growth model is fitted in all cases with a single random component $b_i$ allowing for a tree specific asymptotic circumference

$$y_{ij} = \frac{\beta_1 + b_i}{1 + \exp\left[-(t_j - \beta_2)/\beta_3\right]} + \epsilon_{ij}, \quad i = 1, ..., 5, \quad j = 1, ..., 7, \quad (3)$$
with $\epsilon_{ij} \sim N(0, \sigma^2)$ and $b_i \sim N(0, \sigma_b^2)$ and mutually independent. The matrices $\Sigma$ and $\Psi$ are both diagonal. Here, $\beta_1$ determines the asymptotic circumference, $\beta_2$ is the age at half this value, $\beta_3$ is related to the growth rate and $t_j$ is the time in days since the beginning of the study. The maximum likelihood estimates (MLEs) of the fixed parameters along with standard errors for model (3) are given in Table 1. A plot of the data and model (3) is shown in Figure 1(a).

Table 1: Parameter estimates (standard errors) and log-likelihoods for models estimated in section 3.2 for the orange tree data.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\sigma$</th>
<th>$\sigma_b$</th>
<th>$\sigma_\delta$</th>
<th>$\rho$</th>
<th>log($L$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3)</td>
<td>192.1</td>
<td>727.9</td>
<td>348.1</td>
<td>7.84</td>
<td>31.6</td>
<td>-131.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(15.7)</td>
<td>(35.3)</td>
<td>(27.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td>196.2</td>
<td>748.4</td>
<td>352.9</td>
<td>5.30</td>
<td>32.6</td>
<td>10.5</td>
<td>-125.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(19.4)</td>
<td>(62.3)</td>
<td>(33.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td>217.1</td>
<td>857.5</td>
<td>436.8</td>
<td>0.322</td>
<td>4.79</td>
<td>36.0</td>
<td>-116.79</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(18.1)</td>
<td>(42.0)</td>
<td>(24.5)</td>
<td>(0.038)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4) + (6)</td>
<td>192.4</td>
<td>730.1</td>
<td>348.1</td>
<td>6.12</td>
<td>32.7</td>
<td>12.0</td>
<td>-118.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(19.6)</td>
<td>(63.8)</td>
<td>(34.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5) + (6)</td>
<td>216.2</td>
<td>859.1</td>
<td>437.8</td>
<td>0.330</td>
<td>5.76</td>
<td>36.7</td>
<td>-106.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(17.6)</td>
<td>(30.5)</td>
<td>(21.6)</td>
<td>(0.022)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Plots for model (3) for orange tree data.

A plot of residuals versus time (sampling occasion) shown in Figure 1(b) reveals an unmodeled variation with time as is also noted by Millar (2004). Millar proposes to include a second random component, $b_{2j}$ for the sampling occasion, that is, crossed with the random...
component for trees. He suggests the model

$$y_{ij} = \frac{\beta_1 + b_{1i} + b_{2j}}{1 + \exp\left[-(t_j - \beta_2)/\beta_3\right]} + \epsilon_{ij}$$

(4)

with $$b_{2j} \sim N(0, \sigma_{b_2}^2)$$ and independent of $$b_{1i}$$ and $$\epsilon_{ij}$$, which successfully removes the most significant structure in the residuals. In this model, the effect of the sampling occasion, $$b_{2j}$$ is proportional to the model prediction. This is reasonable during the initial growth period, but unreasonable when the trees reach their asymptotic circumference. Rather, we find it more natural to include $$b_{2j}$$ additively in the exp-term in the denominator in model (3) to make the random effects additive on the logit-scale. This makes the effect of the sampling occasion vanish as the trees approach their asymptotic circumference.

A closer look at the sampling scheme reveals, however, that the apparently random effect of the sampling occasion is caused by a seasonal effect and an irregular sampling pattern. In the residual plot in Figure 1(b), it is seen that all samples are taken either in the spring (April or May) or in the fall (October) and that two periods are missing. We include a categorical seasonal effect, $$\beta_4$$ such that

$$y_{ij} = \beta_1 + b_{1i} + \frac{b_{2j}}{1 + \exp\left[-((t_j - \beta_2)/\beta_3 + s_j\beta_4)\right]} + \epsilon_{ij}$$

(5)

where $$s_j$$ is $$-1/2$$ and $$1/2$$ for samples taken in the spring and fall respectively. The models (4) and (5) still show significant unmodeled serial correlation in the residuals within trees. This may be modelled with a continuous auto-regressive (CAR) process (e.g. Diggle et al., 2002; Pinheiro and Bates, 2000) for the residuals by assuming

$$\text{cov}(\epsilon_{ij}, \epsilon_{i'j'}) = \sigma^2 \exp(-\phi|t_{ij'} - t_j|/(365/2)), \quad \phi \geq 0$$

(6)

so the full covariance matrix is block diagonal with $$\Sigma(\phi, \sigma) = I_5 \otimes \text{cov}(\epsilon_i)$$ where $$\otimes$$ denotes the Kronecker product. The time is scaled so that $$\rho = \exp(-\phi)$$ can be interpreted as the correlation over half a year and therefore roughly between sampling occasions. Model (4) with crossed random effects cannot easily, if at all, be fitted with standard software for NLMMs. We return to the models described above in Section 3 and show how they can all be easily estimated by means of the Laplace approximation and implemented on a
2 The Laplace Approximation

In this section we outline the Laplace approximation to the full $q$-dimensional integral in the marginal likelihood function (2) for easy reference in later sections. For more details, see Wolfinger and Lin (1997).

The Laplace approximation consists of approximating the logarithm of the integrand in the marginal likelihood (2), i.e. the joint log-likelihood

$$h(\theta, b; y) = \log p(y; \theta, b) + \log p(b)$$

by a second-order Taylor expansion,

$$t(\theta, b; y) = h(\theta, \tilde{b}; y) - \frac{1}{2}(b - \tilde{b})^T H(\theta, \tilde{b})(b - \tilde{b})$$

for which the integral has an explicit solution. The expansion is performed around the maximizer of the joint log-likelihood function (conditional mode), i.e. $\tilde{b} = \arg \max_b h(\theta, b; y)$, which gives the best approximation of the integrand (Barndorff-Nielsen and Cox, 1979). The negative Hessian, $H(\theta, \tilde{b}) = f_b^p \Sigma^{-1} f_b^T - f_{bb}^p \Sigma^{-1} (y - f) + \Psi^{-1}|_{b=\tilde{b}}$, can be approximated by

$$D(\theta, \tilde{b}) = f_b^p \Sigma^{-1} f_b^T + \Psi^{-1}|_{b=\tilde{b}},$$

where the second-order term that usually contributes negligibly (Bates and Watts, 1980) has been omitted. This approximation is the expected Hessian similar to the approximation used in the Gauss-Newton nonlinear least-squares and Fisher scoring methods. The Laplace approximation using $D$ rather than $H$ is referred to as the modified Laplace approximation by Pinheiro and Bates (1995) and in its most general form reads

$$l_{LA}(\theta; y) = \log \int_{\mathbb{R}^s} \exp \{t(\theta, b; y)\} \, db$$

$$= h(\theta, \tilde{b}; y) - \frac{1}{2} \log |D(\theta, \tilde{b})/(2\pi)|.$$  

(9)

There are no constraints on how the random effects enter the model function and thus arbi-
trary vector-valued, nested, crossed or partially crossed random effects are accommodated.

If the random effects, \( b \) appear linearly in the model function, \( f \), the Laplace approximation is exact because the second-order Taylor expansion is exact in this case. The modified Laplace approximation (9) is also exact in this case because the second-order term ignored in \( D \) is zero. We will sometimes refer to (9) as the Laplace likelihood because it depends on the particular model whether it is exact or an approximation.

We can view the Laplace likelihood as an approximation of the integrand in the marginal likelihood (2) by a Gaussian curve. The approximation therefore improves as the integrand, i.e. the joint likelihood, gets closer to a Gaussian curve or equivalently as the joint log-likelihood (7) gets closer to quadratic function. Vonesh (1996) shows that as the number of observations increase per random effect, the joint log-likelihood tends to a quadratic function – a fact also supported by the Bayesian central limit theorem (Carlin and Louis, 2000, p.122-124). Because the integral of the joint likelihood is exactly or closely approximated by the integral of an approximating Gaussian curve, we find that the Laplace approximation is a natural approximation for estimation in NLMMs.

3 Model Estimation

This section demonstrates how the Laplace approximation to the marginal likelihood can be used to implement NLMMs on a case-by-case basis with a very limited amount of coding required. This estimation scheme opens up for a great deal of flexibility and enables estimation of a range of models that are not otherwise supported by today’s standard software packages. We describe the computational approach in section 3.1 and illustrate the implementation in section 3.2.

3.1 Computational approach

Our computational approach is based on estimating the parameters of the Laplace likelihood (9) by a general purpose quasi-Newton optimizer, for instance of the BFGS-type (e.g. Nocedal and Wright, 2006). To evaluate the Laplace likelihood (9) for a set of parameters, \( \theta \), two quantities \( \hat{b} \) and \( D(\theta, \hat{b}) \) has to be available. This leads to a nested optimization, since for every evaluation of the Laplace likelihood with a set of parameters, \( \theta \) in the outer optimization, the joint log-likelihood, \( h \) has to be optimized over \( b \) in the inner optimization.
We also use a general purpose quasi-Newton optimizer for the latter task. The only unknown
quantity needed to evaluate $D(\theta, \tilde{b})$ given $\theta$ and $\tilde{b}$ is the Jacobian, $f'_b$ for which we use a
finite difference approximation. Implementation of any NLMM consists of three functions:
The model function, $f$, the joint log-likelihood, $h$ in (7) and the Laplace likelihood in (9).

The starting values in the inner optimization is simply zero; the expectation of the
random effects. Starting values for the regression parameters, $\beta$ are based on plots of the
data or previous fits of other models, potentially fixed effect versions. Starting values for
variance and correlation parameters are qualified guesses based on plots of the data.

At convergence of the outer optimization, we use a finite difference approximation to the
Hessian to obtain the variance-covariance matrix of the parameters.

Any inaccuracies in the estimation of $\tilde{b}$ and $f'_b$ are directly reflected as noise in the
Laplace likelihood. For the gradient based estimation of the model parameters to converge
smoothly, it is therefore important to obtain sufficiently good estimates of these quantities.

The variance parameters are optimized on the scale of the logarithm of the standard
deviation to make the estimation unbounded and to make the log-likelihood surface more
quadratic facilitating faster convergence. Because all terms in the Laplace likelihood (9) are
evaluated on the log-scale, it can be evaluated for any variance parameter arbitrarily close
to the boundary at zero (in finite computer arithmetic). This ensures that the optimization
will proceed smoothly even if the (MLE) is zero. Further, it allows the likelihood to be
profiled with respect to the variance parameters arbitrarily close to zero.

The optimizer \texttt{nlminb} in the base package in \texttt{R} is chosen for the inner and outer op-
timizations. The Jacobian is estimated using the numerical approximation implemented
in \texttt{jacobian} in the \texttt{numDeriv} package (Gilbert, 2009). The \texttt{hessian} function, also from the
\texttt{numDeriv} package, is used to obtain a finite difference estimation of the Hessian at the
convergence of the outer optimization.

The computational approach described above can be optimized with respect to speed
and robustness in a number of respects – generally at the cost of flexibility and more complex
coding. Analytical gradients of the joint log-likelihood with respect to the random effects
can be found via the general expression $h'_b(b) = f'_b\Sigma^{-1}(y - f) - \Psi^{-1}b$ and will increase the
speed of convergence of the inner optimization. The analytical Jacobian will also increase the
speed of computation of the Hessian approximation. A Gauss-Newton type estimation of the
random effects can replace the quasi-Newton optimization of the inner optimization using the gradient and the Hessian, $D$ from above. This often leads to further speed improvements. In models with many random effects, the quasi-Newton estimation in the inner optimization can benefit from memory-limited BFGS updates (Nocedal and Wright, 2006), and the Gauss-Newton estimation can benefit from the use of sparse matrix methods (e.g. Davis, 2006). Lastly, the residual variance can be profiled out of the likelihood thus reducing the dimension of the outer optimization (Pinheiro and Bates, 1995).

3.2 Implementation in R

In this section we show how the models for the orange tree data presented in section 1.1 can be estimated in a few lines of code with the computational approach for the Laplace approximation described in section 3.1. First, we show how the simple model (3) with a single scalar random effect is implemented. Subsequently we show how this code can be altered in only a few places to fit model (4) with crossed random effects, model (5) with a seasonal effect, and how the CAR process in (6) can be allowed for in the latter models.

The model function, $f$ for model (3) is defined as

```r
> f <- function(beta, b) {
  (beta[1] + rep(b[1:5], each = 7))/
  (1 + exp((beta[2] - time)/beta[3]))
}
```

The function returns a vector of the same length as the data with model predictions based on the 3 fixed effects in $\beta$, the 5 random effects in $b$ and the 7 time points in $\text{time}$. The joint negative log-likelihood based on (7) is defined as

```r
> h <- function(b, beta, sigma, sigma.b) {
  -sum(dnorm(x = circumference, mean = f(beta, b),
   sd = sigma, log = TRUE)) -
  sum(dnorm(x = b[1:5], sd = sigma.b, log = TRUE))
}
```

using two vectorized calls to the univariate normal density function $\text{dnorm}$, because the conditional distribution of the observations and the distribution of the random effects are mutually independent normal. This is the negative joint log-likelihood, because standard optimization algorithms by default minimize rather than maximize.

Based on the implementations of the model function and the joint log-likelihood, the Laplace approximation to the marginal log-likelihood $l_{LA}(\theta)$ is implemented as
where the parameters to be estimated are \( \theta = (\beta, \log \sigma, \log \sigma_b) \). The call to \texttt{nlminb} in \texttt{l.LA} performs the inner optimization and computes \( \hat{b} \), and the Hessian, \( D \) is computed as in (8) based on the Jacobian, \( f'_b \).

The maximum likelihood fit of model (3) is obtained by performing the outer optimization with the call

\[ > \texttt{fit <- nlminb(theta0, l.LA)} \]

where the starting values, \( \theta^0 \) are inferred from Figure 1(a). The estimation converges in a few seconds on a standard personal computer to the model fit presented in Table 1. This concludes the estimation of model (3). Next, we show how this code can be changed to estimate model (4) with crossed random effects. We mention all changes to the code apart from updates to the parameter sets passed between functions and similar trivialities. Only small changes to the code are required to estimate a model that is not within reach with standard software for NLMMs. This illustrates the power and flexibility of the proposed estimation scheme.

Model (4) has two crossed random components \( b_1 \) and \( b_2 \) for tree and time and the full vector of random effects is thus \( b = [b_{11}, ..., b_{15}, b_{21}, ..., b_{27}]^T \). The model function \( f \) is modified to include the 7 new random effects for sampling occasion by adding the term \( \text{rep}(b[6:12], 5) \) to \( \text{beta}[1] + \text{rep}(b[1:5], \text{each} = 7) \). To accommodate the additional random effects with standard deviation \( \sigma_{b2} \) in the joint log-likelihood, the function \( h \) is updated by adding \( -\text{sum(dnorm(x = b[6:12], sd = sigma.b2, log = TRUE))} \) to the existing code. The only change to the Laplace likelihood, \texttt{1.LA} is in the adaption of the change
in the covariance matrix for the random effects, $\Psi$ to the Hessian, $D$ in (8); the term $\text{diag}(1/\text{sigma.b}^2, 5)$ is replaced by $\text{diag}(c(\text{rep}(1/\text{sigma.b}^2, 5), \text{rep}(1/\text{sigma.b}^2, 7)))$.

Model (4) is estimated similarly to model (3) and the optimization converges in a matter of seconds to the results shown in Table 1.

While model (4) is a significant improvement to model (3), model (5) with a seasonal effect might be more appropriate than model (4) with a random effect of sampling occasion. To fit model (5), the only change to the previously defined functions; $f$, $h$ and $l.LA$ for model (3) is the addition of the term $\text{beta}[4] \times \text{season}$ in the $\exp$-term in $f$. The estimate of model (5) is also shown in Table 1. Because the likelihood of model (5) is considerably higher than that of model (4) at the same expense of parameters and producing almost exactly the same predictions, we prefer model (5).

The model fit for model (5) is shown in Figure 2(a). By comparing this to the fit of model (3) in Figure 1(a), it appears that model (5) seems to capture the variation between sampling occasions. This is also verified by a plot of residuals versus time in Figure 2(b), where the residuals within sample occasions are now centered around zero and smaller than in Figure 1(b). The plots for model (4) are very similar to those in Figure 2 for model (5).

![Figure 2: Plots for model (5) for orange tree data.](image)

In Figure 2(b), the residuals for each tree have been connected by lines to illustrate that a positive auto-correlation is present. Only small changes to the estimation scheme are required to accommodate any correlation or covariance structure in the residuals. In the
following we will describe how the CAR process in (6) for the within tree residuals can be implemented and included in the estimation of the models (4) and (5). We implement the covariance matrix, $\Sigma$ in (6) as

```r
> Sigma.CAR <- function(phi, sigma) {
    diff <- (time[1:7] - rep(time[1:7], each=7))
    delta.t <- matrix(diff / (365 / 2), nrow = 7, ncol = 7)
    P <- sigma^2 * exp(-phi * abs(delta.t))
    kronecker(diag(5), P)
}
```

where $\delta.t$ is a matrix of time differences and $P$ is $\text{cov}(\epsilon_i)$. To accommodate the CAR process in the residuals in models (4) and (5), the model functions remain as previously described and the joint log-likelihood is defined as

```r
> h <- function(b, beta, sigma, sigma.b, sigma.b2, phi) {
    Sigma <- Sigma.CAR(phi, sigma)
    resid <- circumference - f(beta, b)
    0.5 * (log(det(2*pi*Sigma)) + crossprod(resid, solve(Sigma, resid)) - sum(dnorm(x = b[1:5], sd = sigma.b, log=TRUE)) -
            sum(dnorm(x = b[6:12], sd = sigma.b2, log=TRUE))
}
```

where the notable difference from previously is that the first part of $h$ is now written as the logarithm of a multivariate normal density using the full residual covariance matrix $\Sigma$ (in model (5) the last call to $\text{dnorm}$ concerning $\sigma_{b2}$ is excluded). The term $(y - \mu)^T \Sigma^{-1} (y - \mu)$ in the normal density function is computed using $\text{crossprod}(\text{resid}, \text{solve}($\Sigma$, \text{resid}))$, since this is numerically more stable and more efficient than computing the term directly as defined. The only change to $l.LA$ to accommodate (6) is in the computation of the Hessian, $D$, where $\text{crossprod}(\text{Jac.f})/\text{sig}^2$ is changed to $\text{crossprod}(\text{Jac.f, solve}($\Sigma$, \text{Jac.f}))$.

To make the estimation of the correlation parameter, $\phi$ in the CAR process (6) unbounded, it is optimized on the log-scale. The estimates of models (4) and (5) with the CAR process (6) are shown in Table 1. For both models, the CAR process is a significant improvement with $p$-values $<0.001$ based on likelihood ratio tests. For model (5), the correlation over half a year, and therefore roughly between sampling occasions, is $\hat{\rho} = 0.81$, which is equivalent to the correlation coefficient in a discrete AR(1) model, where account is taken of missing sampling occasions. This corresponds to the strong auto-correlation seen in Figure 2(b) between successive sampling occasions.
4 Graphical Methods for Inference and Validation

In this section we illustrate a number of graphical methods that are useful for inference and validation of a model fit. First, we discuss how to obtain the profile likelihood and corresponding confidence intervals, next we discuss the use of pseudo likelihoods for assessing convergence, and finally we discuss how the accuracy of the Laplace approximation can be assessed when random effects enter nonlinearly in the model function.

The profile likelihood is an inferential tool in its own right, and it can be used to make likelihood based confidence intervals instead of having to rely on the Wald approximation. For a scalar parameter $\theta$, the profile likelihood is defined as $L(\theta) = \max_\eta L(\theta, \eta)$, where $\eta$ are nuisance parameters and $\theta = (\theta, \eta)$. Our approach contains a single loop over the parameter of interest with repeated optimization with respect to the remaining nuisance parameters. The profile likelihood can be interpolated by a spline (e.g. spline in R) to reduce the number of values of $\theta$ for which the likelihood has to be optimized to produce a smooth curve. Figure 3(a) shows the relative profile likelihood for the variance parameter for the random effects of sampling occasion in model (4), and Figure 3(b) shows the relative profile likelihood for the half-year correlation $\rho$ in model (5) with the seasonal effect and the CAR residual structure. The horizontal lines at 0.1465 and 0.03625 define 95% and 99% confidence intervals based on the usual $\chi^2_1$-asymptotics of the likelihood ratio statistic. The profile likelihood confidence bounds can be found by numerically solving for the intersection of the spline function with these threshold (e.g. using uniroot in R). The figures show which values of the parameters are supported by the data and which are of negligible likelihood relative to the MLE. The figures also illustrate the effect of the arguably arbitrary choice of confidence level.

The profile likelihood can be time consuming to compute if the parameter set is large because of the many optimizations that are required. If interest is in assessing convergence, the many optimizations can be avoided by making use of the pseudo (or estimated) likelihood. The pseudo likelihood is given as $L_e(\theta) = L(\theta, \hat{\eta})$ where $\hat{\eta}$ is the MLE of $\eta$. It ignores the uncertainty in the remaining parameters, and there is no general way to use it for frequency calibrated inference, but it can be useful for visually checking that the optimization has converged at the optimum. Figure 4 shows pseudo log-likelihood plots for all parameters in model (4) around their MLE on the scale at which the optimization is performed. The
Figure 3: Relative profile likelihoods for (a) the variance parameter for the random effects of sampling occasion in model (4) and (b) the correlation over half a year in model (5) + (6). The horizontal lines indicate 95% and 99% confidence intervals.

plots indicate proper convergence to an unequivocal optimum. If the pseudo log-likelihood is plotted for a very small range around the MLEs, the plots can also be used to study the accuracy in the evaluation of the Laplace log-likelihood as inaccuracies will show up as noise on the curve. In this way it is found that the error of the log-likelihood surface is on the order $10^{-7}$ to $10^{-10}$ for the models we have considered. This is sufficient for the gradient based estimation of model parameters to be both robust and efficient.

Figure 4: Pseudo log-likelihood profiles for model (4).

The Laplace approximation is exact for all models estimated in Section 3, because the random effects enter linearly in the model functions. Sometimes however, random effects enter nonlinearly and the Laplace likelihood (9) is an approximation to the marginal likeli-
hood (2), and it is of interest to substantiate how accurate the approximation is. We may assess the accuracy of the approximation by graphical means, and to illustrate this, we will use a model that was briefly mentioned in the introduction

\[ y_{ij} = \frac{\beta_1 + b_{1j}}{1 + \exp[-((t_j - \beta_2)/\beta_3 + b_{2j})]} + \epsilon_{ij} \]  

(10)

where \( b_{2j} \) enter additively on the logit-scale and thus nonlinearly in the model function. This model has a log-likelihood of -125.39 which is almost identical to that of model (4), cf. Table 1. Because the two random components in (10) are crossed, the integral defining the marginal likelihood (2) is 12-dimensional. We know that the Laplace approximation is exact in the 5 directions corresponding to the linear random effect, \( b_{11} \). We can examine how good the Laplace approximation to the 12-dimensional integral is in the directions corresponding to the random effects that appear nonlinearly in the model function. This is not a rigid assessment of the accuracy of the entire integral, but intuitively we expect the total error to be small, if the error is negligible in the directions corresponding to the random effects that appear nonlinearly in the model function. Because the random effects in each random component are independent by definition, and because each random effect in one component only depends on the random effects in the other component indirectly through the fixed parameters, the Hessian, \( D \) (and \( H \)) is very close to orthogonal. Therefore, the error of the Laplace approximation will not be notably larger in the directions that we are not examining. The integrand is given by the joint likelihood, and we may plot this as a function of one of the random effects that enter nonlinearly in the model function, while holding the fixed effects and the remaining random effects fixed at their estimates, \( \hat{\theta} \) and \( \tilde{b} \). The approximating Gaussian curve illustrates the Laplace approximation and is based on \( \tilde{b} \) and the appropriate diagonal entry of \( D(\hat{\theta}, \tilde{b}) \). Figure 5 illustrates the joint likelihood (solid line) and the approximating Gaussian curve (dashed line) for the random effects \( b_{24} \) (a) and \( b_{27} \) (b). We expect there to be more information about the random effect \( b_{24} \) than about \( b_{27} \) due to the structure of the logistic curve (\( b_{24} \) is at a sampling occasion, where the slope of the logistic curve is large, and \( b_{27} \) is near the asymptotic circumference, where variations has a smaller influence on the model function, cf. Figure 1(a)). This is reflected in Figure 5 in two respects: 1) The curve is wider for \( b_{27} \) than for \( b_{24} \), 2) the curve for \( b_{24} \) is better approximated by the Gaussian curve than the curve for \( b_{27} \).
Using a general integration function (\texttt{integrate} in base package in R), we evaluated the integrals of the joint likelihoods in Figure 5 with sufficient precision and found that the relative error of the Laplace approximation in these directions are 0.08% and 0.22%. The error in the directions corresponding to the remaining random effects that enter nonlinearly in the model (10) is of similar size, and we conclude that the error of the Laplace approximation is negligible from a statistical perspective for this model.

In model (10), $D$ is an approximation to $H$ due to the nonlinearity of the model function in the random effects. At the parameter estimates the absolute error in using $D$ rather than $H$ in the term $-\log |D/(2\pi)|/2$ in the Laplace approximation (9) is 0.0058 which is irrelevant from a statistical perspective and lends support to the previous remark that the second-order term ignored in $D$ is of negligible magnitude.

## 5 Comparison to Standard Software

The Laplace approximation gives the exact marginal likelihood for model (3) for the orange data, because random effects enter linearly in the model function. The model is an example of a simple NLMM with just one random component and can be handled by all standard software. The model is used to compare the accuracy of the R-based estimation scheme to SAS NLMIXED, NONMEM and \texttt{nlme} and the results are shown in Table 2. Both SAS NLMIXED and NONMEM were used with the Laplace approximation, and, as can be seen from the table, they both agree with the implementation presented in this paper to all reported digits (NONMEM uses an objective function missing a constant term $\log(2\pi)\sum n_i$ (Wang, 2007), which has been corrected for in the table). Also the parameter and std. err.
estimates were found to be very similar. The last row in Table 2 is \texttt{nlme} using Lindstrom and Bates' (1990) alternating method, which is not exact for this model. The approximate log-likelihood at the MLE deviates slightly from the others, which is also the case for the parameter estimates.

<table>
<thead>
<tr>
<th>Software</th>
<th>Log-likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{nlme in R}</td>
<td>-131.5718851</td>
</tr>
<tr>
<td>SAS NLMIXED</td>
<td>-131.57188</td>
</tr>
<tr>
<td>NONMEM</td>
<td>-131.5718</td>
</tr>
<tr>
<td>\texttt{nlme in nlme}</td>
<td>-131.5845</td>
</tr>
</tbody>
</table>

6 Discussion

The presented estimation scheme using the multivariate Laplace approximation offers a very large flexibility in the specification of NLMMs at the cost of only a rather limited amount of coding. It provides an option to fit models, assess convergence and draw inference via profile likelihoods when standard software falls short. Especially models with crossed random effects are not (at least easily) handled by any currently available software package such as NONMEM, SAS NLMIXED, or \texttt{nlme} for R/S-Plus. In this way the approach presented here fills a gap left by standard software for NLMMs. The analysis of the orange tree data presented here shows that the flexibility and ability to estimate, compare and draw inference from various models is of substantial importance to the conclusions of the data analysis. The approximation error of the Laplace approximation appears to be of only minor importance.

Several other methods for approximating the marginal likelihood of NLMMs than that of Laplace have been proposed including Gauss Hermite quadrature (GHQ) (e.g. Davidian and Gallant, 1992), adaptive Gauss Hermite quadrature (AGQ) (e.g. Lui and Pierce, 1994; Rabe-Hesketh et al., 2005), simulation methods and linearization methods. Pinheiro and Bates (1995) compare these methods for models with a single level of grouping and conclude that Laplace and AGQ are the most appealing if one is not content with the linearization method of Lindstrom and Bates (1990). In models with random effects that enter linearly in the model function, the Laplace likelihood is exact, so there is no need to go to lengths with the more computationally demanding AGQ, but AGQ will improve Laplace in models with nonlinear random effects. In models with one level of grouping or nested random effects.
components, the integration problems can typically be held in small dimensions, and AGQ can be computationally feasible, but since the number of evaluations of the joint likelihood increases exponentially with the dimension, the scope of the method is limited. A more sophisticated approach is to integrate over a sparse grid rather than the full grid as proposed by Heiss and Winschel (2008). For models with crossed or partially crossed random effect structures, the dimension of the integral increases linearly with the number of random effects, so sparse grid integration also quickly reaches its feasible limit. As an example, consider model (10) with crossed random effects, where one of the random components enter nonlinearly in the model function. The dimension of the integral is 12, and AGQ with a modest 5 quadrature points would require $5^{12} = 244,140,625$ evaluations of the joint likelihood at each evaluation of the approximation to the marginal likelihood. The sparse grid methods reduce this number substantially so that in 10 and 20 dimensions, the number of points is, respectively, 5,281 and 90,561 (Heiss and Winschel, 2008). For the small orange tree data this would be within range of standard computing power, but if the number of trees is doubled or tripled, this also becomes too inefficient.

Stochastic methods such as simulated likelihood is to some extent applicable to models with crossed or partially crossed random effects, but it also suffers from the curse of dimensionality. As noted by Pinheiro and Bates (1995), the inherent uncertainty in stochastic approximations makes the likelihood hard to optimize. Millar (2004) uses simulated likelihood with 50,000 importance samples and exploiting antithetic variables to estimate model (4). He reports a value of the log-likelihood for model (4) that differ 0.0017 from that of the Laplace likelihood, which is exact for this model with an absolute error less than $10^{-7}$ as discussed in section 4. However, if the log-likelihood is evaluated at the parameter estimates reported by Millar, the actual error of Millar’s simulated likelihood is only 0.0002. Although this difference is irrelevant from an inferential viewpoint, it illustrates the inherent uncertainty in stochastic methods that can hinder optimization of the likelihood.

The estimation times for our implementations are generally longer compared to those of standard software packages, when these are able to estimate the specified models. The most complex model considered here is model (4) with the CAR structure and it takes a few minutes to fit. The remaining models converges in a matter of a few seconds directly to the MLE without any further effort. Larger data sets will inevitably increase estimation
times, but not necessarily the code complexity. In some cases the optimizations of the
computational approach mentioned in section 3.1 might be worth the effort.

In this paper we have illustrated the flexibility of estimation with the multivariate Laplace
approximation in the framework of NLMMs, but this method is applicable for a much larger
class of models. Essentially all mixed models, where the joint likelihood is easily defined can
be accommodated including the important generalized linear mixed models; univariate as
well as multivariate, and also the less common generalized nonlinear mixed models. When
the distribution of the observations is not Gaussian, the Laplace approximation is naturally
less accurate, but by the Bayesian central limit theorem (Carlin and Louis, 2000), the joint
likelihood tends to a normal curve when the number of observations per random effect
increase, so the Laplace approximation can be expected to be fairly accurate when the
information per random effect is not small. In our experience the Laplace approximation is
remarkably accurate for a wide range of models, but further research is needed to address
this topic formally.

Appendix

Table 3: Circumference in millimeters for 5 orange trees reported by Draper and Smith

<table>
<thead>
<tr>
<th>Tree</th>
<th>118$^S$</th>
<th>484$^S$</th>
<th>664$^A$</th>
<th>1004$^A$</th>
<th>1231$^S$</th>
<th>1372$^A$</th>
<th>1582$^S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>58</td>
<td>87</td>
<td>115</td>
<td>120</td>
<td>142</td>
<td>145</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>69</td>
<td>111</td>
<td>156</td>
<td>172</td>
<td>203</td>
<td>203</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>51</td>
<td>75</td>
<td>108</td>
<td>115</td>
<td>139</td>
<td>140</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>62</td>
<td>112</td>
<td>167</td>
<td>179</td>
<td>209</td>
<td>214</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>49</td>
<td>81</td>
<td>125</td>
<td>142</td>
<td>174</td>
<td>177</td>
</tr>
</tbody>
</table>

$^A, ^S$: Autumn, Spring

REFERENCES

Barndorff-Nielsen, O. and D. R. Cox (1979). Edgeworth and saddle-point approximations
279–312.


Appendix H

Reference manual for the R package ordinal

Package ‘ordinal’

January 20, 2012

Type Package

Title Regression Models for Ordinal Data

Version 2012.01-19

Date 2012-01-19

Author Rune Haubo B Christensen

Maintainer Rune Haubo B Christensen <rhbc@imm.dtu.dk>

Imports numDeriv

Depends R (>= 2.13.0), MASS, ucminf, Matrix,

Suggests lme4, nnet, xtable

Description This package implements cumulative link (mixed) models also known as ordered regression models, proportional odds models, proportional hazards models for grouped survival times and ordered logit/probit/... models. Estimation is via maximum likelihood and mixed models are fitted with the Laplace approximation and adaptive Gauss-Hermite quadrature. Multiple random effect terms are allowed and they may be nested, crossed or partially nested/crossed. Restrictions of symmetry and equidistance can be imposed on the thresholds (cut-points). Standard model methods are available (summary, anova, drop-methods, step, confint, predict etc.) in addition to profile methods and slice methods for visualizing the likelihood function and checking convergence.

License GPL (> 2)

Archs i386, x64

R topics documented:

ordinal-package .................................................. 2
addterm.clm2 .................................................... 4
anova.clm2 ....................................................... 5
clm ................................................................. 7
clm.control ....................................................... 10
clm.fit ............................................................ 11
clm2 ............................................................... 13
clm2.control ..................................................... 18
clmm ............................................................... 19
Description

This package facilitates analysis of ordinal (ordered categorical data) via cumulative link models (CLMs) and cumulative link mixed models (CLMMs). Robust and efficient computational methods gives speedy and accurate estimation. A wide range of methods for model fits aids the data analysis.

Details

This package implements cumulative link models and cumulative link mixed models with normally distributed random effects, denoted cumulative link mixed (effects) models. Cumulative link models are also known as ordered regression models, proportional odds models, proportional hazards models for grouped survival times and ordered logit/probit/... models.

Cumulative link models are fitted with `clm` and the main features are:

- A range of standard link functions are available.
- In addition to the standard location (additive) effects, scale (multiplicative) effects are also allowed.
• nominal effects are allowed for any subset of the predictors — these effects are also known as partial proportional odds effects when using the logit link.

• Restrictions can be imposed on the thresholds/cut-points, e.g., symmetry or equidistance.

• A (modified) Newton-Raphson algorithm provides the maximum likelihood estimates of the parameters. The estimation scheme is robust, fast and accurate.

• Rank-deficient designs are identified and unidentified coefficients exposed in print and summary methods as with glm.

• A suite of standard methods are available including anova, add/drop-methods, step, profile, confint.

• A slice method facilitates illustration of the likelihood function and a convergence method summarizes the accuracy of the model estimation.

• The predict method can predict probabilities, response class-predictions and cumulative probabilities, and it provides standard errors and confidence intervals for the predictions.

Cumulative link mixed models are fitted with clmm and the main features are:

• Any number of random effect terms can be included.

• The syntax for the model formula resembles that of lmer

• Nested random effects, crossed random effects and partially nested/crossed random effects are allowed.

• Estimation is via maximum likelihood using the Laplace approximation or adaptive Gauss-Hermite quadrature (one random effect).

• Vector-valued and correlated random effects are not yet implemented.

• Random slopes (random coefficient models) are not yet implemented.

• Estimation employs sparse matrix methods from the Matrix package.

• During model fitting a Newton-Raphson algorithm updates the conditional modes of the random effects a large number of times. The likelihood function is optimized with a general purpose optimizer.

In addition to the reference manual several vignettes describe aspects of the package. Two brief tutorials on clm and clmm introduces the use of these functions. The Primer vignette is a 30-page introduction to cumulative link models.

A major update of the package in August 2011 introduced new and improved implementations of clm and clmm. The old implementations are available with clm2 and clmm2. At the time of writing there is functionality in clm2 and clmm2 not yet available in clm and clmm. This includes flexible link functions (log-gamma and Aranda-Ordaz links) and a profile method for random effect variance parameters in CLMMs. The new implementations are expected to take over the old implementations at some point, hence the latter will eventually be deprecated and defunct.

Author(s)

Rune Haubo B Christensen

Maintainer: Rune Haubo B Christensen <rhbc@imm.dtu.dk>

Examples

```r
## A simple cumulative link model:
data(wine)
fm1 <- clm(rating ~ contact + temp, data=wine)
```
## A simple cumulative link mixed model:
fmm1 <- clmm(rating ~ contact + temp + (1|judge), data=wine)
summary(fmm1)
**Value**

A table of class "anova" containing columns for the change in degrees of freedom, AIC and the likelihood ratio statistic. If `test = "Chisq"` a column also contains the p-value from the Chisq test.

**Author(s)**

Rune Haubo B Christensen

**See Also**

`clm2`, `anova`, `addterm.default` and `dropterm.default`

**Examples**

```r
options(contrasts = c("contr.treatment", "contr.poly"))
data(soup)
mB1 <- clm2(SURENESS ~ PROD + GENDER + SOUPTYPE,
  scale = ~ COLD, data = soup, link = "probit",
  Hess = FALSE)
dropterm(mB1, test = "Chi") # or
dropterm(mB1, which = "location", test = "Chi")
dropterm(mB1, which = "scale", test = "Chi")
addterm(mB1, scope = ~ .^2, test = "Chi", which = "location")
addterm(mB1, scope = ~ . + GENDER + SOUPTYPE,
  test = "Chi", which = "scale")
addterm(mB1, scope = ~ . + AGEGROUP + SOUPFREQ,
  test = "Chi", which = "location")

## Fit model from polr example:
data(housing, package = "MASS")
fM1 <- clm2(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
addterm(fM1, ~ Infl + Type + Cont, test= "Chisq", which = "scale")
dropterm(fM1, test = "Chisq")
```

---

**Description**

Comparison of cumulative link models in likelihood ratio tests. The models may differ by terms in location, scale and nominal formulae, in link, threshold function and random effect structure.

**Usage**

```r
## S3 method for class 'clm2'
anova(object, ..., test = c("Chisq", "none"))
## S3 method for class 'clmm2'
anova(object, ..., test = c("Chisq", "none"))
```
Arguments

object a `clm2` object.
...
... one or more additional `clm2` objects.
test if `test = "none"` the p-value for the likelihood ratio test is suppressed.

Value

The method returns an object of class `Anova` (for printing) and `data.frame` with the following elements

Model character description of the cumulative link models being compared. Location, scale and nominal formulae are separated by "I"s in this order.
Resid.df the residual degrees of freedom
-2logLik twice the negative log likelihood (proportional to the deviance)
Test indication of which models are being compared.
DF the difference in the degrees of freedom in the models being compared, i.e. the degrees of freedom for the chi-squared test.
LR stat. the likelihood ratio statistic.
Pr(Chi) the p-value from the likelihood ratio test. Absent if `test = "none"`.

Author(s)

Rune Haubo B Christensen

See Also

`clm2`, `addterm`, `dropterm` and `anova.default`

Examples

```r
options(contrasts = c("contr.treatment", "contr.poly"))
data(soup)
m1 <- clm2(SURENESS ~ PROD, scale = ~PROD, data = soup,
   link = "logistic")

## anova
anova(m1, update(m1, scale = ~.-PROD))
mN1 <- clm2(SURENESS ~ 1, nominal = ~PROD, data = soup,
   link = "logistic")
anova(m1, mN1)
anova(m1, update(m1, scale = ~.-PROD), mN1)

## Fit model from polr example:
data(housing, package = "MASS")
fm1 <- clm2(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
anova(fm1, update(fm1, scale = ~ Cont))
```
**clm**

*Cumulative Link Models*

**Description**

Fits cumulative link models (CLMs) such as the proportional odds model. The model allows for various link functions and structured thresholds that restricts the thresholds or cut-points to be e.g., equidistant or symmetrically arranged around the central threshold(s). Nominal effects (partial proportional odds with the logit link) are also allowed. A modified Newton algorithm is used to optimize the likelihood function.

**Usage**

```r
clm(formula, scale, nominal, data, weights, start, subset, doFit = TRUE, na.action, contrasts, model = TRUE, control=list(), link = c("logit", "probit", "cloglog", "loglog", "cauchit"), threshold = c("flexible", "symmetric", "equidistant"), ...)
```

**Arguments**

- **formula**
  A formula expression as for regression models, of the form `response ~ predictors`. The response should be a factor (preferably an ordered factor), which will be interpreted as an ordinal response with levels ordered as in the factor. The model must have an intercept: attempts to remove one will lead to a warning and will be ignored. An offset may be used. See the documentation of `formula` for other details.

- **scale**
  An optional formula expression, of the form `~ predictors`, i.e. with an empty left hand side. An offset may be used. Variables included here will have multiplicative effects and can be interpreted as effects on the scale (or dispersion) of a latent distribution.

- **nominal**
  An optional formula of the form `~ predictors`, i.e. with an empty left hand side. The effects of the predictors in this formula are assumed to nominal - this corresponds to the so-called partial proportional odds (with the logit link).

- **data**
  An optional data frame in which to interpret the variables occurring in the formulas.

- **weights**
  Optional case weights in fitting. Defaults to 1.

- **start**
  Initial values for the parameters in the format `c(alpha, beta, zeta)`, where alpha are the threshold parameters (adjusted for nominal effects), beta are the regression parameters and zeta are the scale parameters.

- **subset**
  Expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.

- **doFit**
  Logical for whether the model should be fit or the model environment should be returned.

- **na.action**
  A function to filter missing data. Applies to terms in all three formulae.

- **contrasts**
  A list of contrasts to be used for some or all of the factors appearing as variables in the model formula.

- **model**
  Logical for whether the model frame should be part of the returned object.

- **control**
  A list of control parameters passed on to `clm.control`. 


link function, i.e., the type of location-scale distribution assumed for the latent distribution. The default "logit" link gives the proportional odds model.

threshold specifies a potential structure for the thresholds (cut-points). "flexible" provides the standard unstructured thresholds, "symmetric" restricts the distance between the thresholds to be symmetric around the central one or two thresholds for odd or equal numbers or thresholds respectively, and "equidistant" restricts the distance between consecutive thresholds to be of the same size.

Additional arguments are passed on to `clm.control`.

Details

This is a new (as of August 2011) improved implementation of CLMs. The old implementation is available in `clm2`, but will probably be removed at some point.

There are methods for the standard model-fitting functions, including `summary`, `anova`, `model.frame`, `model.matrix`, `drop1`, `dropterm`, `step`, `stepAIC`, `extractAIC`, `AIC`, `coef`, `nobs`, `profile`, `confint`, `vcov` and `slice`.

Value

If `doFit = FALSE` the result is an environment representing the model ready to be optimized. If `doFit = TRUE` the result is an object of class "clm2" with the components listed below.

Note that some components are only present if `scale` and `nominal` are used.

- **alpha**: a vector of threshold parameters.
- **beta**: a vector of regression parameters.
- **zeta**: a vector of scale regression parameters.
- **coefficients**: a vector of coefficients of the form `c(alpha, beta, zeta)`
- **aliased**: list of length 3 or less with components `alpha`, `beta` and `zeta` each being logical vectors containing alias information for the parameters of the same names.
- **start**: the parameter values at which the optimization has started. An attribute `start.iter` gives the number of iterations to obtain starting values for models where `scale` is specified or where the `cauchit` link is chosen.
- **fitted.values**: the fitted probabilities.
- **logLik**: the value of the log-likelihood at the estimated optimum.
- **edf**: the estimated degrees of freedom, i.e., the number of parameters in the model fit.
- **nobs**: the number of observations counted as `sum(weights)`.
- **n**: the number of observations counted as `nrow(X)`, where `X` is the design matrix.
- **gradient**: a vector of gradients for the coefficients at the estimated optimum.
- **maxGradient**: the maximum absolute gradient, i.e., `max(abs(gradient))`.
- **Hessian**: the Hessian matrix for the parameters at the estimated optimum.
- **convergence**: convergence code where 0 indicates successful convergence, 1 indicates the iteration limit was reached before convergence and 2 indicates the step factor was reduced below minimum before convergence was reached.
- **link**: character, the link function used.
- **threshold**: character, the threshold structure used.
- **call**: the matched call.
contrasts (where relevant) the contrasts used for the formula part of the model.
S.contrasts (where relevant) the contrasts used for the scale part of the model.
nom.contrasts (where relevant) the contrasts used for the nominal part of the model.
terms the terms object for the formula part.
S.terms (where relevant) the terms object for the scale part.
nom.terms (where relevant) the terms object for the nominal part.
xlevels (where relevant) a record of the levels of the factors used in fitting for the formula part.
S.xlevels (where relevant) a record of the levels of the factors used in fitting for the scale part.
nom.xlevels (where relevant) a record of the levels of the factors used in fitting for the nominal part.
tJac the transpose of the Jacobian for the threshold structure.
y.levels the levels of the response variable.
info a table of basic model information for printing.
model if requested (the default), the model.frame containing variables from formula, scale and nominal parts.
na.action (where relevant) information returned by model.frame on the special handling of NAs.

Note

Note that the number of iterations shown in the info table by print and summary methods can in some cases exceed the maximum number of iterations. This can happen because in a model where scale is specified, the iterations to obtain starting values are added to the main iterations.

Author(s)

Rune Haubo B Christensen

Examples

data(wine)
fm1 <- clm(rating ~ temp * contact, data = wine)
fm1 ## print method
summary(fm1)
fm2 <- update(fm1, ~.-temp:contact)
anova(fm1, fm2)
dropterm(fm1, test = "Chi")
drop1(fm1, test = "Chi")
add1(fm1, ~.+judge, test = "Chi")
addterm(fm1, ~.+judge, test = "Chi")
fm2 <- step(fm1)
summary(fm2)
fm3 <- stepAIC(fm1)
summary(fm3)
coef(fm1)
vcov(fm1)
AIC(fm1)
extractAIC(fm1)
logLik(fm1)
fitted(fm1)

confint(fm1) ## type = "profile"
confint(fm1, type = "Wald")
pr1 <- profile(fm1)
confint(pr1)

## other link functions:
fm4.lgt <- update(fm1, link = "logit") ## default
fm4.prt <- update(fm1, link = "probit")
fm4.ll <- update(fm1, link = "loglog")
fm4.cll <- update(fm1, link = "cloglog")
fm4.cct <- update(fm1, link = "cauchit")
anova(fm4.lgt, fm4.prt, fm4.ll, fm4.cll, fm4.cct)

## structured thresholds:
fm5 <- update(fm1, threshold = "symmetric")
fm6 <- update(fm1, threshold = "equidistant")
anova(fm1, fm5, fm6)

## the slice methods:
slice.fm1 <- slice(fm1)
par(mfrow = c(3, 3))
plot(slice.fm1)

## Example from MASS::polr:
data(housing, package = "MASS")
fm1 <- clm(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
summary(fm1)

## Another example:
data(soup, package = "ordinal")
fm.soup <- clm(SURENESS ~ PRODID, data = soup)
summary(fm.soup)
Description
Set control parameters for cumulative link models

Usage
clm.control(method = c("Newton", "model.frame", "ucminf", "nlminb", "optim"), ..., trace = 0, maxIter = 100, gradTol = 1e-06, maxLineIter = 15)

Arguments

  method  "Newton" fits the model by maximum likelihood and "model.frame" cause clm to return the model.frame.
  trace   numerical, if > 0 information is printed about and during the optimization process. Defaults to 0.
  maxIter the maximum number of Newton-Raphson iterations. Defaults to 100.
  gradTol the maximum absolute gradient. This is the termination criterion and defaults to 1e-6.
  maxLineIter the maximum number of step halfings allowed if a Newton(-Raphson) step over shoots. Defaults to 10.
  ...    control arguments parsed on to ucminf, nlminb or optim.

Value
a list of control parameters.

Author(s)
Rune Haubo B Christensen

See Also
clm

clm.fit  

Fit Cumulative Link Models

Description
A direct fitter of cumulative link models.

Usage
clm.fit(y, X, S, N, weights = rep(1, nrow(X)), offset = rep(0, nrow(X)), S.offset = rep(0, nrow(X)), control = list(), start, link = c("logit", "probit", "cloglog", "loglog", "cauchit"), threshold = c("flexible", "symmetric", "equidistant"))
clm.fit

Arguments

- **y**: the response variable; a factor, preferably and ordered factor.
- **X, S, N**: optional design matrices for the regression parameters, scale parameters and nominal parameters respectively.
- **weights**: optional case weights.
- **offset**: an optional offset.
- **S.offset**: an optional offset for the scale part of the model.
- **control**: a list of control parameters, optionally a call to `clm.control`.
- **start**: an optional list of starting values of the form `c(alpha, beta, zeta)` for the thresholds and nominal effects (`alpha`), regression parameters (`beta`) and scale parameters (`zeta`).
- **link**: the link function.
- **threshold**: the threshold structure, see further at `clm`.

Details

This function does almost the same thing that `clm` does: it fits a cumulative link model. The main differences are that `clm.fit` does not setup design matrices from formulae and only does minimal post processing after parameter estimation.

Compared to `clm`, `clm.fit` does little to warn the user of any problems with data or model. However, `clm.fit` will attempt to identify column rank deficient designs. Any unidentified parameters are indicated in the aliased component of the fit.

In contrast to `clm`, `clm.fit` allows non-positive weights.

Value

A list with the following components:

- **par**: the maximum likelihood estimate of the model parameters on the form \( c(\alpha, \beta, \zeta) \) for the thresholds and nominal effects (`alpha`), regression parameters (`beta`) and scale parameters (`zeta`).
- **gradient**: gradient of the log-likelihood function for the parameters at the optimum.
- **Hessian**: the Hessian of the parameters at the optimum.
- **logLik**: the value of the log-likelihood function at the optimum.
- **convergence**: a convergence code: 1: successful convergence, 2: iteration limit reached, 3: step factor reduced below minimum.
- **message**: convergence message.
- **maxGradient**: the maximum absolute gradient: \( \max(\text{abs}(\text{gradient})) \).
- **niter**: depends on the optimizer. For the default Newton optimizer it is a vector of length 2: the number of iteration and the number of step halvings respectively.
- **fitted**: the fitted values (probabilities).
- **coef.names**: a list of vectors of names of the coefficients as inferred from the column names of the design matrices.
- **aliased**: a list of logical vectors; TRUE is a parameter is not identified due to column rank deficiency in the design matrices and FALSE otherwise.
**clm2**

**Author(s)**

Rune Haubo B Christensen

**See Also**

clm

**Examples**

```r
## A simple example:
data(wine)
fml <- clm(rating ~ contact + temp, data=wine)
summary(fml)
## get the model frame containing y and X:
mf1 <- update(fml, method="model.frame")
clm.fit(mf1$y, mf1$X)
```

---

**clm2**  
Cumulative link models

**Description**

A new improved implementation of CLMs is available in clm.

Fits cumulative link models with an additive model for the location and a multiplicative model for the scale. The function allows for structured thresholds. A popular special case of a CLM is the proportional odds model. In addition to the standard link functions, two flexible link functions, "Arandar-Ordaz" and "log-gamma" are available, where an extra link function parameter provides additional flexibility. A subset of the predictors can be allowed to have nominal rather than ordinal effects. This has been termed "partial proportional odds" when the link is the logistic.

**Usage**

```r
clm2(location, scale, nominal, data, weights, start, subset, na.action, contrasts, Hess = TRUE, model, link = c("logistic", "probit", "cloglog", "loglog", "cauchit", "Aranda-Ordaz", "log-gamma"), lambda, dofit = TRUE, control, threshold = c("flexible", "symmetric", "equidistant"), ...)
```

**Arguments**

- **location**  
  a formula expression as for regression models, of the form `response ~ predictors`. The response should be a factor (preferably an ordered factor), which will be interpreted as an ordinal response with levels ordered as in the factor. The model must have an intercept: attempts to remove one will lead to a warning and will be ignored. An offset may be used. See the documentation of `formula` for other details.

- **scale**  
  a optional formula expression as for the location part, of the form `~ predictors`, i.e. with an empty left hand side. An offset may be used. See the documentation of `formula` for other details.
nominal an optional formula of the form ~ predictors, i.e. with an empty left hand side. The effects of the predictors in this formula are assumed to nominal.

data an optional data frame in which to interpret the variables occurring in the formulas.

weights optional case weights in fitting. Defaults to 1.

start initial values for the parameters in the format c(alpha, beta, log(zeta), lambda).

subset expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.

na.action a function to filter missing data. Applies to terms in all three formulae.

contrasts a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.

Hess logical for whether the Hessian (the inverse of the observed information matrix) should be computed. Use Hess = TRUE if you intend to call summary or vcov on the fit and Hess = FALSE in all other instances to save computing time. The argument is ignored if method = "Newton" where the Hessian is always computed and returned. Defaults to TRUE.

model logical for whether the model frames should be part of the returned object.

link link function, i.e. the type of location-scale distribution assumed for the latent distribution. The Aranda-Ordaz and log-gamma links add additional flexibility with a link function parameter, lambda. The Aranda-Ordaz link (Aranda-Ordaz, 1983) equals the logistic link, when lambda = 1 and approaches the loglog link when lambda approaches zero. The log-gamma link (Genter and Farewell, 1985) equals the loglog link when lambda = 1, the probit link when lambda = 0 and the cloglog link when lambda = -1.

lambda numerical scalar: the link function parameter. Used in combination with link Aranda-Ordaz or log-gamma and otherwise ignored. If lambda is specified, the model is estimated with lambda fixed at this value and otherwise lambda is estimated by ML. For Aranda-Ordaz lambda has to be positive; > 1e-5 for numerical reasons.

doFit logical for whether the model should be fit or the model environment should be returned.

control a call to clm2.control.

threshold specifies a potential structure for the thresholds (cut-points). "flexible" provides the standard unstructured thresholds, "symmetric" restricts the distance between the thresholds to be symmetric around the central one or two thresholds for odd or equal numbers or thresholds respectively, and "equidistant" restricts the distance between consecutive thresholds to the same value.

Details

There are methods for the standard model-fitting functions, including summary, vcov, predict, anova, logLik, profile, plot.profile, confint, update, dropterm, addterm, and an extractAIC method.

The design of the implementation is inspired by an idea proposed by Douglas Bates in the talk "Exploiting sparsity in model matrices" presented at the DSC conference in Copenhagen, July 14.
clm2

2009. Basically an environment is set up with all the information needed to optimize the likelihood function. Extractor functions are then used to get the value of likelihood at current or given parameter values and to extract current values of the parameters. All computations are performed inside the environment and relevant variables are updated during the fitting process. After optimizer termination relevant variables are extracted from the environment and the remaining are discarded.

Some aspects of clm2, for instance, how starting values are obtained, and of the associated methods are inspired by polr from package MASS.

**Value**

If doFit = FALSE the result is an environment representing the model ready to be optimized. If doFit = TRUE the result is an object of class "clm2" with the following components:

- **beta** the parameter estimates of the location part.
- **zeta** the parameter estimates of the scale part on the log scale; the scale parameter estimates on the original scale are given by exp(zeta).
- **Alpha** vector or matrix of the threshold parameters.
- **Theta** vector or matrix of the thresholds.
- **xi** vector of threshold parameters, which, given a threshold function (e.g. "equidistant"), and possible nominal effects define the class boundaries, Theta.
- **lambda** the value of lambda if lambda is supplied or estimated, otherwise missing.
- **coefficients** the coefficients of the intercepts (theta), the location (beta), the scale (zeta), and the link function parameter (lambda).
- **df.residual** the number of residual degrees of freedoms, calculated using the weights.
- **fitted.values** vector of fitted values for each observation. An observation here is each of the scalar elements of the multinomial table and not a multinomial vector.
- **convergence** TRUE if the gradient based convergence criterion is met and FALSE otherwise.
- **gradient** vector of gradients for all the parameters at termination of the optimizer.
- **optRes** list with results from the optimizer. The contents of the list depends on the choice of optimizer.
- **logLik** the log likelihood of the model at optimizer termination.
- **Hessian** if the model was fitted with Hess = TRUE, this is the Hessian matrix of the parameters at the optimum.
- **scale** model.frame for the scale model.
- **location** model.frame for the location model.
- **nominal** model.frame for the nominal model.
- **edf** the (effective) number of degrees of freedom used by the model.
- **start** the starting values.
- **convTol** convergence tolerance for the maximum absolute gradient of the parameters at termination of the optimizer.
- **method** character, the optimizer.
- **y** the response variable.
- **lev** the names of the levels of the response variable.
- **nobs** the (effective) number of observations, calculated as the sum of the weights.
- **threshold** character, the threshold function used in the model.
estimLambda 1 if lambda is estimated in one of the flexible link functions and 0 otherwise.

link character, the link function used in the model.
call the matched call.
contrasts contrasts applied to terms in location and scale models.
na.action the function used to filter missing data.

Author(s)

Rune Haubo B Christensen

References


Examples

```r
options(contrasts = c("contr.treatment", "contr.poly"))
data(soup)

## More manageable data set:
(tab26 <- with(soup, table("Product" = PROD, "Response" = SURENESS)))
dimnames(tab26)[[2]] <- c("Sure", "Not Sure", "Guess", "Guess", "Not Sure", "Sure")
dat26 <- expand.grid(sureness = as.factor(1:6), prod = c("Ref", "Test"))
dat26$wghts <- c(t(tab26))
m1 <- clm2(sureness ~ prod, scale = ~prod, data = dat26, 
weights = wghts, link = "logistic")

## link functions:
m2 <- update(m1, link = "probit")
m3 <- update(m1, link = "cloglog")
m4 <- update(m1, link = "loglog")
m5 <- update(m1, link = "cauchit", start = coef(m1))
m6 <- update(m1, link = "Aranda-Ordaz", lambda = 1)
m7 <- update(m1, link = "Aranda-Ordaz")
m8 <- update(m1, link = "log-gamma", lambda = 1)
m9 <- update(m1, link = "log-gamma")

## nominal effects:
```
\begin{verbatim}
mN1 <- clm2(sureness ~ 1, nominal = ~ prod, data = dat26, weights = wghts, link = "logistic")
anova(m1, mN1)

## optimizer / method:
update(m1, scale = ~ 1, method = "Newton")
update(m1, scale = ~ 1, method = "nlminb")
update(m1, scale = ~ 1, method = "optim")

## threshold functions
mT1 <- update(m1, threshold = "symmetric")
mT2 <- update(m1, threshold = "equidistant")
anova(m1, mT1, mT2)

## Extend example from polr in package MASS:
## Fit model from polr example:
data(housing, package = "MASS")
fm1 <- clm2(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
fm1
summary(fm1)

## With probit link:
summary(update(fm1, link = "probit"))
## Allow scale to depend on Cont-variable
summary(fm2 <- update(fm1, scale =~ Cont))
anova(fm1, fm2)

## which seems to improve the fit

#########################################################################
## It is possible to fit multinomial models (i.e. with nominal
## effects) as the following example shows:
library(nnet)
(hous1.mu <- multinom(Sat ~ 1, weights = Freq, data = housing))
(hous1.clm <- clm2(Sat ~ 1, weights = Freq, data = housing))

## It is the same likelihood:
all.equal(logLik(hous1.mu), logLik(hous1.clm))

## and the same fitted values:
fitHous.mu <-
t(fitted(hous1.mu))[t(col(fitted(hous1.mu)) == unclass(housing$Sat))]
all.equal(fitted(hous1.clm), fitHous.mu)

## The coefficients of multinom can be retrieved from the clm2-object
## by:
Pi <- diff(c(0, plogis(hous1.clm$xi), 1))
log(Pi[2:3]/Pi[1])

## A larger model with explanatory variables:
(hous.mu <- multinom(Sat ~ Infl + Type + Cont, weights = Freq, data = housing))
(hous.clm <- clm2(Sat ~ 1, nominal = ~ Infl + Type + Cont, weights = Freq, data = housing))

## Almost the same likelihood:
all.equal(logLik(hous.mu), logLik(hous.clm))

## And almost the same fitted values:
\end{verbatim}
fitHous.mu <-
 t(fitted(hous.mu))[t(col(fitted(hous.mu)) == unclass(housing$Sat))]
all.equal(fitted(hous.clm), fitHous.mu)
all.equal(round(fitted(hous.clm), 5), round(fitHous.mu), 5)

---

clm2.control

*Set control parameters for cumulative link models*

**Description**

Set control parameters for cumulative link models

**Usage**

```r
clm2.control(method = c("ucminf", "Newton", "nlminb", "optim",
"model.frame"), ..., convTol = 1e-4,
trace = 0, maxIter = 100, gradTol = 1e-5,
maxLineIter = 10)
```

**Arguments**

- `method`: the optimizer used to maximize the likelihood function. "Newton" only works for models without scale, structured thresholds and flexible link functions, but is considerably faster than the other optimizers when applicable. `model.frame` simply returns a list of model frames with the location, scale and nominal model frames. "optim" uses the "BFGS" method.
- `...`: control arguments passed on to the chosen optimizer; see `ucminf`, `optim`, and `nlminb` for details.
- `convTol`: convergence criterion on the size of the maximum absolute gradient.
- `trace`: numerical, if > 0 information is printed about and during the optimization process. Defaults to 0.
- `maxIter`: the maximum number of Newton-Raphson iterations. Defaults to 100.
- `gradTol`: the maximum absolute gradient. This is the termination criterion and defaults to 1e-5.
- `maxLineIter`: the maximum number of step halfings allowed if a Newton(-Raphson) step over shoots. Defaults to 10.

**Value**

a list of control parameters.

**Author(s)**

Rune Haubo B Christensen

**See Also**

`clm2`
Cumulative Link Mixed Models

Description
Fits Cumulative Link Mixed Models with one or more random effects via the Laplace approximation.

Usage
clmm(formula, data, weights, start, subset, na.action, contrasts, Hess = TRUE, model = TRUE, link = c("logit", "probit", "cloglog", "loglog", "cauchit"), doFit = TRUE, control = list(), nAGQ = 1L, threshold = c("flexible", "symmetric", "equidistant"), ...)

Arguments
- formula: a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. The vertical bar character "|" separates an expression for a model matrix and a grouping factor.
- data: an optional data frame in which to interpret the variables occurring in the formula.
- weights: optional case weights in fitting. Defaults to 1.
- start: optional initial values for the parameters in the format c(alpha, beta, tau), where alpha are the threshold parameters, beta are the fixed regression parameters and tau are variance parameters for the random effects on the log scale.
- subset: expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.
- na.action: a function to filter missing data.
- contrasts: a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.
- Hess: logical for whether the Hessian (the inverse of the observed information matrix) should be computed. Use Hess = TRUE if you intend to call summary or vcov on the fit and Hess = FALSE in all other instances to save computing time.
- model: logical for whether the model frames should be part of the returned object.
- link: link function, i.e. the type of location-scale distribution assumed for the latent distribution. The default "logit" link gives the proportional odds mixed model.
- doFit: logical for whether the model should be fit or the model environment should be returned.
- control: a call to clmm.control
- nAGQ: integer; the number of quadrature points to use in the adaptive Gauss-Hermite quadrature approximation to the likelihood function. The default (1) gives the Laplace approximation. Higher values generally provide higher precision at the expense of longer computation times, and values between 5 and 10 generally provide accurate maximum likelihood estimates. Negative values give the non-adaptive Gauss-Hermite quadrature approximation, which is generally faster but...
less accurate than the adaptive version. See the references for further details. Quadrature methods are only available with a single random effects term; the Laplace approximation is always available.

threshold specifies a potential structure for the thresholds (cut-points). "flexible" provides the standard unstructured thresholds, "symmetric" restricts the distance between the thresholds to be symmetric around the central one or two thresholds for odd or equal numbers or thresholds respectively, and "equidistant" restricts the distance between consecutive thresholds to the same value.

... additional arguments are passed on to clm.control.

Details

This is a new (as of August 2011) improved implementation of CLMMs. The old implementation is available in clmm2. Some features are not yet available in clmm; for instance quadrature methods, scale effects, nominal effects and flexible link functions are currently only available in clmm2. clmm is expected to take over clmm2 at some point. There are standard print, summary and anova methods implemented for "clmm" objects.

Value

a list containing

alpha threshold parameters.
beta fixed effect regression parameters.
stDev standard deviation of the random effect terms.
tau log(stDev) - the scale at which the log-likelihood function is optimized.
coefficients the estimated model parameters = c(alpha, beta, tau).
Hessian Hessian of the model coefficients.
edf the estimated degrees of freedom used by the model = length(coefficients).
nobs sum(weights).
n length(y).
fitted.values fitted values evaluated with the random effects at their conditional modes.
df.residual residual degrees of freedom; length(y) - sum(weights)
tJac Jacobian of the threshold function corresponding to the mapping from standard flexible thresholds to those used in the model.
terms the terms object for the fixed effects.
contrasts contrasts applied to the fixed model terms.
na.action the function used to filter missing data.
call the matched call.
logLik value of the log-likelihood function for the model at the optimum.
Niter number of Newton iterations in the inner loop update of the conditional modes of the random effects.
optRes list of results from the optimizer.
ranef list of the conditional modes of the random effects.
condVar list of the conditional variance of the random effects at their conditional modes.
Author(s)
Rune Haubo B Christensen

Examples

```r
## Get test data:
data(soup)
## Cumulative link mixed model with two random terms:
mm1 <- clmm(SURENESS ~ PROD + (1|RESP) + (1|RESP:PROD), data = soup,
             link = "probit", threshold = "equidistant")
mm1
summary(mm1)
## test random effect:
mm2 <- clmm(SURENESS ~ PROD + (1|RESP), data = soup,
             link = "probit", threshold = "equidistant")
anova(mm1, mm2)
```

clmm.control

Set control parameters for cumulative link mixed models

Description
Set control parameters for cumulative link mixed models

Usage

```r
clmm.control(method = c("ucminf", "model.frame"), ..., trace = 0,
              maxIter = 50, gradTol = 1e-4, maxLineIter = 50, innerCtrl =
              c("warnOnly", "noWarn", "giveError"))
```

Arguments

- `method`: the optimizer used to maximize the marginal likelihood function.
- `...`: control arguments passed on to the optimizer; see `ucminf` for details.
- `trace`: numerical, if > 0 information is printed about and during the outer optimization process, if < 0 information is also printed about the inner optimization process. Defaults to 0.
- `maxIter`: the maximum number of Newton updates of the inner optimization. 50.
- `gradTol`: the maximum absolute gradient of the inner optimization.
- `maxLineIter`: the maximum number of step halfings allowed if a Newton(-Raphson) step over shoots during the inner optimization.
- `innerCtrl`: the use of warnings/errors if the inner optimization fails to converge.

Value

a list of control parameters
Author(s)
Rune Haubo B Christensen

See Also
clmm

Description
Fits cumulative link mixed models, i.e. cumulative link models with random effects via the Laplace approximation or the standard and the adaptive Gauss-Hermite quadrature approximation. The functionality in clm2 is also implemented here. Currently only a single random term is allowed in the location-part of the model.

A new implementation is available in clmm that allows for more than one random effect.

Usage
clmm2(location, scale, nominal, random, data, weights, start, subset, na.action, contrasts, Hess = FALSE, model = TRUE, sdFixed, link = c("logistic", "probit", "cloglog", "loglog", "cauchit", "Aranda-Ordaz", "log-gamma"), lambda, doFit = TRUE, control, nAGQ = 1, threshold = c("flexible", "symmetric", "equidistant"), ...)

Arguments
location as in clm2.
scale as in clm2.
nominal as in clm2.
random a factor for the random effects in the location-part of the model.
data as in clm2.
weights as in clm2.
start initial values for the parameters in the format c(alpha, beta, log(zeta), lambda, log(stDev)) where stDev is the standard deviation of the random effects.
subset as in clm2.
na.action as in clm2.
contrasts as in clm2.
Hess logical for whether the Hessian (the inverse of the observed information matrix) should be computed. Use Hess = TRUE if you intend to call summary or vcov on the fit and Hess = FALSE in all other instances to save computing time.
model as in clm2.
sdFixed If sdFixed is specified (a positive scalar), a model is fitted where the standard deviation for the random term is fixed at the value of sdFixed. If sdFixed is left unspecified, the standard deviation of the random term is estimated from data.
link as in `clm2`.
lambda as in `clm2`.
doFit as in `clm2` although it can also be one of `c("no", "R", "C")`, where "R" use the R-implementation for fitting, "C" (default) use C-implementation for fitting and "no" behaves as FALSE and returns the environment.
control a call to `clm2.control`.
threshold as in `clm2`.
nAGQ the number of quadrature points to be used in the adaptive Gauss-Hermite quadrature approximation to the marginal likelihood. Defaults to 1 which leads to the Laplace approximation. An odd number of quadrature points is encouraged and 3, 5 or 7 are usually enough to achieve high precision. Negative values give the standard, i.e. non-adaptive Gauss-Hermite quadrature.

... additional arguments are passed on to `clm2.control` and possibly further on to the optimizer, which can lead to surprising error or warning messages when mistyping arguments etc.

Details

There are methods for the standard model-fitting functions, including `summary`, `vcov`, `profile`, `plot.profile`, `confint`, `anova`, `logLik`, `predict` and an `extractAIC` method.

A Newton scheme is used to obtain the conditional modes of the random effects for Laplace and AGQ approximations, and a non-linear optimization is performed over the fixed parameter set to get the maximum likelihood estimates. The Newton scheme uses the observed Hessian rather than the expected as is done in e.g. `glmer`, so results from the Laplace approximation for binomial fits should in general be more precise - particularly for other links than the "logistic".

Core parts of the function are implemented in C-code for speed.

The function calls `clm2` to up an environment and to get starting values.

Value

If `doFit = FALSE` the result is an environment representing the model ready to be optimized. If `doFit = TRUE` the result is an object of class "clmm2" with the following components:

- `stDev` the standard deviation of the random effects.
- `Niter` the total number of iterations in the Newton updates of the conditional modes of the random effects.
- `grFac` the grouping factor defining the random effects.
- `nAGQ` the number of quadrature points used in the adaptive Gauss-Hermite Quadrature approximation to the marginal likelihood.
- `ranef` the conditional modes of the random effects, sometimes referred to as "random effect estimates".
- `condVar` the conditional variances of the random effects at their conditional modes.
- `beta` the parameter estimates of the location part.
- `zeta` the parameter estimates of the scale part on the log scale; the scale parameter estimates on the original scale are given by `exp(zeta)`.
- `Alpha` vector or matrix of the threshold parameters.
- `Theta` vector or matrix of the thresholds.
vector of threshold parameters, which, given a threshold function (e.g. "equidistant"),
and possible nominal effects define the class boundaries, Theta.

the value of lambda if lambda is supplied or estimated, otherwise missing.

the coefficients of the intercepts (theta), the location (beta), the scale (zeta),
and the link function parameter (lambda).

the number of residual degrees of freedoms, calculated using the weights.

vector of fitted values conditional on the values of the random effects. Use
predict to get the fitted values for a random effect of zero. An observation
here is taken to be each of the scalar elements of the multinomial table and not
a multinomial vector.

TRUE if the optimizer terminates without error and FALSE otherwise.

vector of gradients for the unit-variance random effects at their conditional modes.

list with results from the optimizer. The contents of the list depends on the
choice of optimizer.

the log likelihood of the model at optimizer termination.

if the model was fitted with Hess = TRUE, this is the Hessian matrix of the
parameters at the optimum.

model.frame for the scale model.

model.frame for the location model.

model.frame for the nominal model.

the (effective) number of degrees of freedom used by the model.

the starting values.

character, the optimizer.

the response variable.

the names of the levels of the response variable.

the (effective) number of observations, calculated as the sum of the weights.

character, the threshold function used in the model.

1 if lambda is estimated in one of the flexible link functions and 0 otherwise.

character, the link function used in the model.

the matched call.

contrasts applied to terms in location and scale models.

the function used to filter missing data.

Rune Haubo B Christensen

Examples

options(contrasts = c("contr.treatment", "contr.poly"))
data(soup)

## More manageable data set:
dat <- subset(soup, as.numeric(as.character(RESP)) <= 24)
dat$RESP <- dat$RESP[drop=TRUE]

m1 <- clmm2(SURENESS ~ PROD, random = RESP, data = dat, link="probit",
            Hess = TRUE, method="ucminf", threshold = "symmetric")

m1
summary(m1)
logLik(m1)
vcov(m1)
extractAIC(m1)
anova(m1, update(m1, location = SURENESS ~ 1, Hess = FALSE))
anova(m1, update(m1, random = NULL))

## Use adaptive Gauss-Hermite quadrature rather than the Laplace
## approximation:
update(m1, Hess = FALSE, nAGQ = 3)

## Use standard Gauss-Hermite quadrature:
update(m1, Hess = FALSE, nAGQ = -7)

###########################################
## Binomial example with data from the lme4-package:
data(cbpp, package = "lme4")
cbpp2 <- rbind(cbpp[,-(2:3)], cbpp[,-(2:3)])
cbpp2 <- within(cbpp2, {
  incidence <- as.factor(rep(0:1, each=nrow(cbpp)))
  freq <- with(cbpp, c(incidence, size - incidence))
})

## Fit with Laplace approximation:
fml <- clmm2(incidence ~ period, random = herd, weights = freq,
             data = cbpp2, Hess = 1)
summary(fml)

## Fit with the adaptive Gauss-Hermite quadrature approximation:
fm2 <- clmm2(incidence ~ period, random = herd, weights = freq,
             data = cbpp2, Hess = 1, nAGQ = 7)
summary(fm2)

clmm2.control

Set control parameters for cumulative link mixed models

Description

Set control parameters for cumulative link mixed models
Usage

clmm2.control(method = c("ucminf", "nlminb", "model.frame"), ...,
trace = 0, maxIter = 50, gradTol = 1e-4,
maxLineIter = 50,
innerCtrl = c("warnOnly", "noWarn", "giveError"))

Arguments

method the optimizer used to maximize the marginal likelihood function.
...
control arguments passed on to the chosen optimizer; see ucminf, optim, and
nlminb for details.
trace numerical, if > 0 information is printed about and during the outer optimization
process, if < 0 information is also printed about the inner optimization process. Defaults to 0.
maxIter the maximum number of Newton updates of the inner optimization. 50.
gradTol the maximum absolute gradient of the inner optimization.
maxLineIter the maximum number of step halvings allowed if a Newton(-Raphson) step over
shoots during the inner optimization.
innerCtrl the use of warnings/errors if the inner optimization fails to converge.

Details

When the default optimizer, ucminf is used, the default values of that optimizers control options
are changed to grtol = 1e-5 and grad = "central".

Value

a list of control parameters.

Author(s)

Rune Haubo B Christensen

See Also

clmm2

Description

Confidence intervals and profile likelihoods for parameters in cumulative link models

Computes confidence intervals from the profiled likelihood for one or more parameters in a cumulative link model, or plots the profile likelihood.
Usage

```r
## S3 method for class 'clm'
confint(object, parm, level = 0.95,
  type = c("profile", "Wald"), trace = FALSE, ...)

## S3 method for class 'profile.clm'
confint(object, parm = seq_len(nprofiles),
  level = 0.95, ...)

## S3 method for class 'clm'
profile(fitted, which.beta = seq_len(nbeta),
  which.zeta = seq_len(nzeta), alpha = /zero.noslash.1,
  max.steps = 5, nsteps = 8, trace = FALSE, step.warn = 5,
  control = list(), ...)

## S3 method for class 'profile.clm'
plot(x, which.par = seq_len(nprofiles),
  level = c(0.95, 0.99), Log = FALSE, relative = TRUE, root = FALSE, approx = root, n = 1e3,
  ask = prod(par("mfcol")) < length(which.par) && dev.interactive(),
  fig = TRUE, ylim = NULL)
```

Arguments

- **object, fitted, x**: a fitted `clm` object or a `profile.clm` object.
- **parm, which.par, which.beta, which.zeta**: a numeric or character vector indicating which regression coefficients should be profiled. By default all coefficients are profiled. Ignored for `confint.clm` where all parameters are considered.
- **level**: the confidence level. For the `plot` method a vector of levels for which horizontal lines should be drawn.
- **type**: the type of confidence interval.
- **trace**: if `trace` is `TRUE` or positive, information about progress is printed.
- **Log**: should the profile likelihood be plotted on the log-scale?
- **relative**: should the relative or the absolute likelihood be plotted?
- **root**: should the (approximately linear) likelihood root statistic be plotted?
- **approx**: should the Gaussian or quadratic approximation to the (log) likelihood be included?
- **fig**: should the profile likelihood be plotted?
- **ask**: logical; if `TRUE`, the user is asked before each plot, see `par(ask=.)`.
- **n**: the no. points used in the spline interpolation of the profile likelihood.
- **ylim**: overrules default y-limits on the plot of the profile likelihood.
- **alpha**: The likelihood is profiled in the 100*(1-alpha)% confidence region as determined by the profile likelihood.
- **control**: a list of control parameters for `clm`. Possibly use `clm.control` to set these.
- **max.steps**: the maximum number of profiling steps in each direction for each parameter.
nsteps the (approximate) number of steps to take in each direction of the profile for each parameter. The step length is determined accordingly assuming a quadratic approximation to the log-likelihood function. The actual number of steps will often be close to nsteps, but will deviate when the log-likelihood functions is irregular.

step.warn a warning is issued if the number of steps in each direction (up or down) for a parameter is less than step.warn. If few steps are taken, the profile will be unreliable and derived confidence intervals will be inaccurate.

... additional arguments to be parsed on to methods.

Details

These confint methods call the appropriate profile method, then finds the confidence intervals by interpolation of the profile traces. If the profile object is already available, this should be used as the main argument rather than the fitted model object itself.

Value

confint: A matrix with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2 in % (by default 2.5% and 97.5%).

plot.profile.clm invisibly returns the profile object, i.e., a list of data.frames with an lroot component for the likelihood root statistic and a matrix par.vals with values of the parameters.

Author(s)

Rune Haubo B Christensen

See Also

profile and confint

Examples

data(wine)
fm1 <- clm(rating ~ temp * contact, data = wine)

## Accurate profile likelihood confidence intervals compared to the
## conventional Wald intervals:
confint(fm1) ## type = "profile"
confint(fm1, type = "Wald")
pr1 <- profile(fm1)
confint(pr1)

## plotting the profiles:
par(mfrow = c(2, 2))
plot(pr1, root = TRUE) ## check for linearity
par(mfrow = c(2, 2))
plot(pr1)
par(mfrow = c(2, 2))
plot(pr1, approx = TRUE)
par(mfrow = c(2, 2))
plot(pr1, Log = TRUE)
par(mfrow = c(2, 2))
plot(pr1, Log = TRUE, relative = FALSE)
## Not likely to be useful but allowed for completeness:
par(mfrow = c(2, 2))
plot(pr1, Log = FALSE, relative = FALSE)

## Example from polr in package MASS:
## Fit model from polr example:
data(housing, package = "MASS")
fm1 <- clm(Sat ~ Infl + Type + Cont, weights = Freq,
data = housing)
pr1 <- profile(fm1)
confint(pr1)
par(mfrow=c(2,2))
plot(pr1)

---

### confint.clm2

Confidence intervals and profile likelihoods for parameters in cumulative link models

#### Description

Computes confidence intervals from the profiled likelihood for one or more parameters in a fitted cumulative link model, or plots the profile likelihood function.

#### Usage

```
## S3 method for class 'clm2'
confint(object, parm, level = 0.95, whichL = seq_len(p),
        whichS = seq_len(k), lambda = TRUE, trace = 0, ...)

## S3 method for class 'profile.clm2'
confint(object, parm = seq_along(Pnames), level = 0.95, ...)

## S3 method for class 'clm2'
profile(fitted, whichL = seq_len(p), whichS = seq_len(k),
        lambda = TRUE, alpha = 0.01, maxSteps = 50, delta = LrootMax/10,
        trace = 0, stepWarn = 8, ...)

## S3 method for class 'profile.clm2'
plot(x, parm = seq_along(Pnames), level = c(0.95, 0.99),
     Log = FALSE, relative = TRUE, fig = TRUE, n = 1e3, ..., ylim = NULL)
```

#### Arguments

- **object**: a fitted clm2 object or a profile.clm2 object.
- **fitted**: a fitted clm2 object.
- **x**: a profile.clm2 object.
parm not used in confint.clm2.
For confint.profile.clm2: a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
For plot.profile.clm2: a specification of which parameters the profile likelihood are to be plotted for, either a vector of numbers or a vector of names. If missing, all parameters are considered.

level the confidence level required.

whichL a specification of which location parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all location parameters are considered.

whichS a specification of which scale parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all scale parameters are considered.

lambda logical. Should profile or confidence intervals be computed for the link function parameter? Only used when one of the flexible link functions are used; see the link-argument in clm2.

trace logical. Should profiling be traced?

alpha Determines the range of profiling. By default the likelihood is profiled in the 99% confidence interval region as determined by the profile likelihood.

maxSteps the maximum number of profiling steps in each direction (up and down) for each parameter.

delta the length of profiling steps. To some extent this parameter determines the degree of accuracy of the profile likelihood in that smaller values, i.e. smaller steps gives a higher accuracy. Note however that a spline interpolation is used when constructing confidence intervals so fairly long steps can provide high accuracy.

stepWarn a warning is issued if the no. steps in each direction (up or down) for a parameter is less than stepWarn (defaults to 8 steps) because this indicates an unreliable profile.

Log should the profile likelihood be plotted on the log-scale?

relative should the relative or the absolute likelihood be plotted?

fig should the profile likelihood be plotted?

n the no. points used in the spline interpolation of the profile likelihood.

ylim overrules default y-limits on the plot of the profile likelihood.

... additional argument(s) for methods including range (for the hidden function profileLambda) that sets the range of values of lambda at which the likelihood should be profiled for this parameter.

Details

These confint methods call the appropriate profile method, then finds the confidence intervals by interpolation of the profile traces. If the profile object is already available, this should be used as the main argument rather than the fitted model object itself.

In plot.profile.clm2: at least one of Log and relative arguments have to be TRUE.
Value

confint: A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2 in % (by default 2.5% and 97.5%). The parameter names are preceded with "loc." or "sca." to indicate whether the confidence interval applies to a location or a scale parameter.

plot.profile.clm2 invisibly returns the profile object.

Author(s)

Rune Haubo B Christensen

See Also

profile and confint

Examples

options(contrasts = c("contr.treatment", "contr.poly"))
data(soup)

## More manageable data set:
(tab26 <- with(soup, table("Product" = PROD, "Response" = SURENESS)))
dimnames(tab26)[[2]] <- c("Sure", "Not Sure", "Guess", "Guess", "Not Sure", "Sure")
dat26 <- expand.grid(sureness = as.factor(1:6), prod = c("Ref", "Test"))
dat26$wghts <- c(t(tab26))
m1 <- clm2(sureness ~ prod, scale = ~prod, data = dat26,
weights = wghts, link = "logistic")

## profile
pr1 <- profile(m1)
par(mfrow = c(2, 2))
plot(pr1)
m9 <- update(m1, link = "log-gamma")
pr9 <- profile(m9, whichL = numeric(0), whichS = numeric(0))
par(mfrow = c(1, 1))
plot(pr9)

plot(pr9, Log=TRUE, relative = TRUE)
plot(pr9, Log=TRUE, relative = TRUE, ylim = c(-4, 0))
plot(pr9, Log=TRUE, relative = FALSE)

## confint
confint(pr9)
confint(pr1)

## Extend example from polr in package MASS:
## Fit model from polr example:
data(housing, package = "MASS")
fm1 <- clm2(Sat ~ Infl + Type + Cont, scale = ~ Cont, weights = Freq,
data = housing)

pr1 <- profile(fm1)
confint(pr1)
convergence

par(mfrow=c(2,2))
plot(pr1)

convergence Check convergence of cumulative link models

Description

Check the accuracy of the parameter estimates of cumulative link models. The number of correct
decimals and number of significant digits is given for the maximum likelihood estimates of the
parameters in a cumulative link model fitted with clm.

Usage

convergence(object, ...)

## S3 method for class 'clm'
convergence(object, digits = max(3, getOption("digits") - 3), ...)

Arguments

object for the clm method an object of class "clm", i.e., the result of a call to clm.
digits the number of digits in the printed table.
... arguments to a from methods. Not used by the clm method.

Details

The number of correct decimals is defined as...
The number of significant digits is defined as ...
The number of correct decimals and the number of significant digits are determined from the nu-
merical errors in the parameter estimates. The numerical errors are determined from the Method
Independent Error Theorem (Elden et al, 2004) and is based on the Newton step evaluated at con-
vergence.

Value

Convergence information. In particular a table where the Error column gives the numerical error in
the parameter estimates. These numbers express how far the parameter estimates in the fitted model
are from the true maximum likelihood estimates for this model. The Cor.Dec gives the number of
correct decimals with which the the parameters are determined and the Sig.Dig gives the number
of significant digits with which the parameters are determined.
The number denoted logLik.error is the error in the value of log-likelihood in the fitted model
at the parameter values of that fit. An accurate determination of the log-likelihood is essential for
accurate likelihood ratio tests in model comparison.

Author(s)

Rune Haubo B Christensen
drop.coef

References

Examples

```r
## Simple model:
data(wine)
fml <- clm(rating ~ contact + temp, data=wine)
summary(fml)
convergence(fml)
```

---

### Insure Full Rank Design Matrix

**Description**

Coefficients (columns) are dropped from a design matrix to insure that it has full rank.

**Usage**

```r
drop.coef(X, silent = FALSE)
```

**Arguments**

- **X**: a design matrix, e.g., the result of `model.matrix` possibly of less than full column rank, i.e., with redundant parameters. Works for `ncol(X) >= 0` and `nrow(X) >= 0`.
- **silent**: should a message not be issued if `X` is column rank deficient?

**Details**

Redundant columns of the design matrix are identified with the LINPACK implementation of the `qr` decomposition and removed. The returned design matrix will have `qr(X)$rank` columns.

**Value**

The design matrix `X` without redundant columns.

**Author(s)**

Rune Haubo B Christensen

**See Also**

`qr` and `lm`
Examples

data(soup, package = "ordinal")
X <- model.matrix(~ PRODID * DAY, data = soup)
ncol(X)
newX <- dropcoef(X)
ncol(newX)

## Essentially this is being computed:
qr.X <- qr(X, tol = 1e-7, LAPACK = FALSE)
newX <- X[, qr.X$pivot[1:qr.X$rank], drop = FALSE]
## is newX of full column rank?
ncol(newX) == qr(newX)$rank
## the number of columns being dropped:
ncol(X) - ncol(newX)

---

**gfun**

*Gradients of common densities*

**Description**

Gradients of common density functions in their standard forms, i.e., with zero location (mean) and unit scale. These are implemented in C for speed and care is taken that the correct results are provided for the argument being `NA`, `NaN`, `Inf`, `-Inf` or just extremely small or large.

**Usage**

gnorm(x)
glogis(x)
gcauchy(x)

**Arguments**

x numeric vector of quantiles.

**Details**

The gradients are given by:

- **gnorm**: If \( f(x) \) is the normal density with mean 0 and spread 1, then the gradient is
  \[
  f'(x) = -xf(x)
  \]

- **glogis**: If \( f(x) \) is the logistic density with mean 0 and scale 1, then the gradient is
  \[
  f'(x) = 2\exp(-x)^2(1 + \exp(-x))^{-3} - \exp(-x)(1 + \exp(-x))^{-2}
  \]
• pcauchy: If \( f(x) = \frac{\pi(1 + x^2)^2}{2} \) is the cauchy density with mean 0 and scale 1, then the
gradient is
\[
f'(x) = -2x\frac{\pi(1 + x^2)^2}{2}
\]
These gradients are used in the Newton-Raphson algorithms in fitting cumulative link models with
\texttt{clm} and cumulative link mixed models with \texttt{clmm}.

\textbf{Value}

a numeric vector of gradients.

\textbf{Author(s)}

Rune Haubo B Christensen

\textbf{See Also}

Gradients of densities are also implemented for the extreme value distribution (\texttt{gumbel}) and the the
log-gamma distribution (\texttt{log-gamma}).

\textbf{Examples}

\begin{verbatim}
x <- -5:5
gnorm(x)
glogis(x)
gcauchy(x)
\end{verbatim}

\begin{verbatim}
gumbel(x)
\end{verbatim}

\textbf{Description}

Density, distribution function, quantile function, random generation, and gradient of density of the
extreme value (maximum and minimum) distributions. The Gumbel distribution is also known as
the extreme value maximum distribution, the double-exponential distribution and the log-Weibull
distribution.

\textbf{Usage}

\begin{verbatim}
dgumbel(x, location = 0, scale = 1, log = FALSE, max = TRUE)
pgumbel(q, location = 0, scale = 1, lower.tail = TRUE, max = TRUE)
qgumbel(p, location = 0, scale = 1, lower.tail = TRUE, max = TRUE)
rgumbel(n, location = 0, scale = 1, max = TRUE)

ggumbel(x, max = TRUE)
\end{verbatim}
Arguments

- **x,q**: numeric vector of quantiles.
- **p**: vector of probabilities.
- **n**: number of observations.
- **location**: numeric scalar.
- **scale**: numeric scalar.
- **lower.tail**: logical; if `TRUE` (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
- **log**: logical; if `TRUE`, probabilities $p$ are given as log($p$).
- **max**: distribution for extreme maxima (default) or minima? The default corresponds to the standard right-skew Gumbel distribution.

Details

dgumbel, pgumbel and ggumbel are implemented in C for speed and care is taken that 'correct' results are provided for values of NA, NaN, Inf, -Inf or just extremely small or large.

See the 'Primer' vignette for the definition of the Gumbel distribution and its relation to the log-log and complementary-log-log link used in cumulative link models. See the examples for numerical relations between the max and min variants.

The distribution functions, densities and gradients are used in the Newton-Raphson algorithms in fitting cumulative link models with `clm` and cumulative link mixed models with `clmm`.

Value

- `pgumbel` gives the distribution function, `dgumbel` gives the density, `ggumbel` gives the gradient of the density, `qgumbel` is the quantile function, and `rgumbel` generates random deviates.

Author(s)

Rune Haubo B Christensen

References

[wikipedia.org/wiki/Gumbel_distribution](http://wikipedia.org/wiki/Gumbel_distribution)

See Also

Gradients of densities are also implemented for the normal, logistic, cauchy, cf. `gfun` and the log-gamma distribution, cf. `lgamma`.

Examples

```r
## Illustrating the symmetry of the distribution functions:
pgumbel(5) == 1 - pgumbel(-5, max=FALSE) ## TRUE
dgumbel(5) == dgumbel(-5, max=FALSE) ## TRUE
ggumbel(5) == -ggumbel(-5, max=FALSE) ## TRUE

## More examples:
x <- -5:5
(pp <- pgumbel(x))
```
income

qgumbel(pp)
dgumbel(x)
ggumbel(x)

(ppp <- pgumbel(x, max=FALSE))
## Observe lack of precision:
qgumbel(ppp, max=FALSE)
dgumbel(x, max=FALSE)
ggumbel(x, max=FALSE)

## random deviates:
set.seed(1)
(r1 <- rgumbel(10))
set.seed(1)
r2 <- -rgumbel(10, max = FALSE)
all(r1 == r2) ## TRUE

income

Income distribution (percentages) in the Northeast US

Description

Usage
income

Format
year year.
pct percentage of population in income class per year.
income income groups. The unit is thousands of constant (1973) US dollars.

Source
Data are adopted from McCullagh (1980).

References

Examples

data(income)

print(income)

## Convenient table:
The log-gamma distribution

Description

Density, distribution function and gradient of density for the log-gamma distribution. These are implemented in C for speed and care is taken that the correct results are provided for values of NA, NaN, Inf, -Inf or just extremely small or large values.

The log-gamma is a flexible location-scale distribution on the real line with an extra parameter, \( \lambda \). For \( \lambda = 0 \) the distribution equals the normal or Gaussian distribution, and for \( \lambda \) equal to 1 and -1, the Gumbel minimum and maximum distributions are obtained.

Usage

\[
\text{plgamma}(q, \lambda, \text{lower.tail} = \text{TRUE})
\]

\[
\text{dlgamma}(x, \lambda, \log = \text{FALSE})
\]

\[
\text{glgamma}(x, \lambda)
\]

Arguments

\( x, q \) numeric vector of quantiles.

\( \lambda \) numerical scalar

\( \text{lower.tail} \) logical; if \( \text{TRUE} \) (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

\( \log \) logical; if \( \text{TRUE} \), probabilities \p are given as \( \log(p) \).
Details

If $\lambda < 0$ the distribution is right skew, if $\lambda = 0$ the distribution is symmetric (and equals the normal distribution), and if $\lambda > 0$ the distribution is left skew.

The log-gamma distribution function is defined as ... pending.

The density and gradient of the density are defined as ... pending.

These distribution functions, densities and gradients are used in the Newton-Raphson algorithms in fitting cumulative link models with clm2 and cumulative link mixed models with clmm2 using the log-gamma link.

Value

plgamma gives the distribution function, dlgamma gives the density and glgamma gives the gradient of the density.

Author(s)

Rune Haubo B Christensen

References


See Also

Gradients of densities are also implemented for the normal, logistic, cauchy, cf. gfun and the Gumbel distribution, cf. gumbel.

Examples

```r
## Illustrating the link to other distribution functions:
x <- -5:5
plgamma(x, lambda = 0) == pnorm(x)
all.equal(plgamma(x, lambda = -1), pgumbel(x)) ## TRUE, but:
plgamma(x, lambda = -1) == pgumbel(x)
plgamma(x, lambda = 1) == pgumbel(x, max = FALSE)

dlgamma(x, lambda = 0) == dnorm(x)
dlgamma(x, lambda = -1) == dgumbel(x)
dlgamma(x, lambda = 1) == dgumbel(x, max = FALSE)

glgamma(x, lambda = 0) == gnorm(x)
all.equal(glgamma(x, lambda = -1), ggumbel(x)) ## TRUE, but:
glgamma(x, lambda = -1) == ggumbel(x)
all.equal(glgamma(x, lambda = 1), ggumbel(x, max = FALSE)) ## TRUE, but:
glgamma(x, lambda = 1) == ggumbel(x, max = FALSE)

## There is a loss of accuracy, but the difference is very small:
glgamma(x, lambda = 1) - ggumbel(x, max = FALSE)

## More examples:
x <- -5:5
plgamma(x, lambda = .5)
dlgamma(x, lambda = .5)
```

predict.clm

## S3 method for class 'clm'
predict(object, newdata, se.fit = FALSE, interval = FALSE,
      level = 0.95, type = c("prob", "class", "cum.prob", "linear.predictor"),
      na.action = na.pass, ...)

### Arguments

- `object`: a fitted object of class inheriting from `clm`.
- `newdata`: optionally, a data frame in which to look for variables with which to predict. Note that all predictor variables should be present having the same names as the variables used to fit the model. If the response variable is present in `newdata` predictions are obtained for the levels of the response as given by `newdata`. If the response variable is omitted from `newdata` predictions are obtained for all levels of the response variable for each of the rows of `newdata`.
- `se.fit`: should standard errors of the predictions be provided? Not applicable and ignored when `type = "class"`.
- `interval`: should confidence intervals for the predictions be provided? Not applicable and ignored when `type = "class"`.
- `level`: the confidence level.
- `type`: the type of predictions. "prob" gives probabilities, "class" gives predicted response class membership defined as highest probability prediction. "cum.prob" gives cumulative probabilities (see details) and "linear.predictor" gives predictions on the scale of the linear predictor.
- `na.action`: function determining what should be done with missing values in `newdata`. The default is to predict `NA`.
- `...`: further arguments passed to or from other methods.

### Details

If `newdata` is supplied and the response variable is omitted, then predictions, standard errors and intervals are matrices rather than vectors with the same number of rows as `newdata` and with one column for each response class. If `type = "class"` predictions are always a vector.

If `newdata` is omitted the predictions are based on the data used for the fit. In that case, the way missing values in the original fit are handled is determined by the `na.action` argument of that fit. If `na.action = na.omit` omitted cases will not appear in the residuals, whereas if `na.action = na.omit`
predict.clm

na.exclude they will appear (in predictions, standard errors or interval limits), with residual value NA. See also napredict.

If type = "cum.prob" or type = "linear.predictor" there will be two sets of predictions, standard errors and intervals; one for j and one for j-1 in the usual notation.

If newdata is supplied and the response variable is omitted, then predict.clm returns much the same thing as predict.polr (matrices of predictions). Similarly, if type = "class".

If the fit is rank-deficient, some of the columns of the design matrix will have been dropped. Prediction from such a fit only makes sense if newdata is contained in the same subspace as the original data. That cannot be checked accurately, so a warning is issued (cf. predict.lm).

Value

A list containing the following components

fit predictions or fitted values if newdata is not supplied.
se.fit if se.fit=TRUE standard errors of the predictions otherwise NULL.
upr, lwr if interval=TRUE lower and upper confidence limits.

Author(s)

Rune Haubo B Christensen

See Also

clm, clmm.

Examples

## simple model:
data(wine)
fm1 <- clm(rating ~ contact + temp, data=wine)
summary(fm1)

## Fitted values with standard errors and confidence intervals:
predict(fm1, se.fit=TRUE, interval=TRUE) # type="prob"
## class predictions for the observations:
predict(fm1, type="class")
newData <- expand.grid(temp = c("cold", "warm"),
                       contact = c("no", "yes"))

## Predicted probabilities in all five response categories for each of
## the four cases in newData:
predict(fm1, newdata=newData, type="prob")
## now include standard errors and intervals:
predict(fm1, newdata=newData, se.fit=TRUE, interval=TRUE, type="prob")
predict.clm2  
**Predict Method for CLM fits**

**Description**

Obtains predictions from a cumulative link (mixed) model.

**Usage**

```r
## S3 method for class 'clm2'
predict(object, newdata, ...)
```

**Arguments**

- `object`: a fitted object of class inheriting from clm2 including clmm2 objects.
- `newdata`: optionally, a data frame in which to look for variables with which to predict. Observe that the response variable should also be present.
- `...`: further arguments passed to or from other methods.

**Details**

This method does not duplicate the behavior of `predict.polr` in package MASS which produces a matrix instead of a vector of predictions. The behavior of `predict.polr` can be mimicked as shown in the examples.

If `newdata` is not supplied, the fitted values are obtained. For clmm2 fits this means predictions that are controlled for the observed value of the random effects. If the predictions for a random effect of zero, i.e., an average 'subject', are wanted, the same data used to fit the model should be supplied in the `newdata` argument. For clm2 fits those two sets of predictions are identical.

**Value**

A vector of predicted probabilities.

**Author(s)**

Rune Haubo B Christensen

**See Also**

clm2, clmm2.

**Examples**

```r
options(contrasts = c("contr.treatment", "contr.poly"))
data(soup)

## More manageable data set for less voluminous printing:
(tab26 <- with(soup, table("Product" = PROD, "Response" = SURENESS)))
dimnames(tab26)[[2]] <- c("Sure", "Not Sure", "Guess", "Guess", "Not Sure", "Sure")
dat26 <- expand.grid(sureness = as.factor(1:6), prod = c("Ref", "Test"))
```
dat26$wghts <- c(t(tab26))
dat26
m1 <- clm2(sureness ~ prod, scale = ~prod, data = dat26,
weights = wghts, link = "logistic")
predict(m1)

mN1 <- clm2(sureness ~ 1, nominal = ~prod, data = dat26,
weights = wghts)
predict(mN1)
predict(update(m1, scale = ~.-prod))

## Fit model from polr example:
data(housing, package = "MASS")
fm1 <- clm2(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
predict(fm1)

#########################################################################
## Mimicking the behavior of predict.polr:
set.seed(123)
nlev <- 3
y <- gl(nlev, 5)
x <- as.numeric(y) + rnorm(15)
fm.clm <- clm2(y ~ x)
fm.polr <- polr(y ~ x)

## The equivalent of predict.polr(object, type = "probs"):
(pmat.polr <- predict(fm.polr, type = "probs"))
ndat <- expand.grid(y = gl(nlev,1), x = x)
(pmat.clm <- matrix(predict(fm.clm, newdata = ndat, ncol=nlev,
byrow = TRUE))
all.equal(c(pmat.clm), c(pmat.polr), tol = 1e-5) # TRUE

## The equivalent of predict.polr(object, type = "class"):
(class.polr <- predict(fm.polr))
(class.clm <- factor(apply(pmat.clm, 1, which.max)))
all.equal(class.clm, class.polr) ## TRUE

profile.clmm2

Confidence intervals and profile likelihoods for the standard
deviation for the random term in cumulative link mixed models

Description

Computes confidence intervals from the profiled likelihood for the standard deviation for the random
term in a fitted cumulative link mixed model, or plots the associated profile likelihood function.

Usage

## S3 method for class 'profile.clmm2'
confint(object, parm = seq_along(Pnames), level = 0.95, ...)
## S3 method for class 'clmm2'
profile(fitted, alpha = 0.01, range, nSteps = 20, trace = 1, ...)

## S3 method for class 'profile.clmm2'
plot(x, parm = seq_along(Pnames), level = c(0.95, 0.99),
    Log = FALSE, relative = TRUE, fig = TRUE, n = 1e3, ..., ylim = NULL)

Arguments

- **object**
  - a fitted profile.clmm2 object.

- **fitted**
  - a fitted clmm2 object.

- **x**
  - a profile.clmm2 object.

- **parm**
  - For confint.profile.clmm2: a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered. Currently only "stDev" or 1 are supported.

  For plot.profile.clmm2: a specification of which parameters the profile likelihood are to be plotted for, either a vector of numbers or a vector of names. If missing, all parameters are considered. Currently only "stDev" or 1 are supported.

- **level**
  - the confidence level required. Observe that the model has to be profiled in the appropriate region; otherwise the limits are NA.

- **trace**
  - logical. Should profiling be traced? Defaults to TRUE due to the time consuming nature of the computation.

- **alpha**
  - Determines the range of profiling. By default the likelihood is profiled approximately in the 99% confidence interval region as determined by the Wald approximation. This is usually sufficient for 95% profile likelihood confidence limits.

- **range**
  - if range is specified, this overrules the range computation based on alpha. range should be all positive and stDev is profiled in range(range).

- **nSteps**
  - the number of points at which to profile the likelihood function. This determines the resolution and accuracy of the profile likelihood function; higher values gives a higher resolution, but also longer computation times.

- **Log**
  - should the profile likelihood be plotted on the log-scale?

- **relative**
  - should the relative or the absolute likelihood be plotted?

- **fig**
  - should the profile likelihood be plotted?

- **n**
  - the no. points used in the spline interpolation of the profile likelihood for plotting.

- **ylim**
  - overrules default y-limits on the plot of the profile likelihood.

- **...**
  - additional argument(s), e.g. graphical parameters for the plot method.

Details

A confint.clmm2 method deliberately does not exist due to the time consuming nature of the computations. The user is required to compute the profile object first and then call confint on the profile object to obtain profile likelihood confidence intervals.

In plot.profile.clm2: at least one of Log and relative arguments have to be TRUE.
slice

Value

confint: A matrix with columns giving lower and upper confidence limits. These will be labelled as (1-level)/2 and 1 - (1-level)/2 in % (by default 2.5% and 97.5%).

plot.profile.clm2 invisibly returns the profile object.

Author(s)

Rune Haubo B Christensen

See Also

profile and confint

Examples

options(contrasts = c("contr.treatment", "contr.poly"))

data(cbpp, package = "lme4")

cbpp2 <- rbind(cbpp[,-(2:3)], cbpp[,-(2:3)])

cbpp2 <- within(cbpp2, {
  incidence <- as.factor(rep(0:1, each=nrow(cbpp)))
  freq <- with(cbpp, c(incidence, size - incidence))
})

## Fit with Laplace approximation:
fm1 <- clmm2(incidence ~ period, random = herd, weights = freq,
               data = cbpp2, Hess = 1)

pr.fm1 <- profile(fm1)
confint(pr.fm1)

par(mfrow = c(2,2))
plot(pr.fm1)
plot(pr.fm1, Log=TRUE, relative = TRUE)
plot(pr.fm1, Log=TRUE, relative = FALSE)

slice

Slice the likelihood of a clm

Description

Slice likelihood and plot the slice. This is useful for illustrating the likelihood surface around the MLE (maximum likelihood estimate) and provides graphics to substantiate (non-)convergence of a model fit. Also, the closeness of a quadratic approximation to the log-likelihood function can be inspected for relevant parameters. A slice is considerably less computationally demanding than a profile.
Usage

slice(object, ...)  
## S3 method for class 'clm'
slice(object, parm = seq_along(par), lambda = 3,  
      grid = 100, quad.approx = TRUE, ...)

## S3 method for class 'slice.clm'
plot(x, parm = seq_along(x),
     type = c("quadratic", "linear"), plot.mle = TRUE,
     ask = prod(par("mfcol")) < length(parm) && dev.interactive(), ...)

Arguments

object for the clm method an object of class "clm", i.e., the result of a call to clm.
x a slice.clm object, i.e., the result of slice(clm.object).
parm for slice.clm a numeric or character vector indexing parameters, for plot.slice.clm only a numeric vector is accepted. By default all parameters are selected.
lambda the number of curvature units on each side of the MLE the slice should cover.
grid the number of values at which to compute the log-likelihood for each parameter.
quad.approx compute and include the quadratic approximation to the log-likelihood function?
type "quadratic" plots the log-likelihood function which is approximately quadratic, and "linear" plots the signed square root of the log-likelihood function which is approximately linear.
plot.mle include a vertical line at the MLE (maximum likelihood estimate) when type = "quadratic"? Ignored for type = "linear".
ask logical; if TRUE, the user is asked before each plot, see par(ask=).
... further arguments to plot.default for the plot method. Not used in the slice method.

Value

The slice method returns a list of data.frames with one data.frame for each parameter slice. Each data.frame contains in the first column the values of the parameter and in the second column the values of the (positive) log-likelihood "logLik". A third column is present if quad.approx = TRUE and contains the corresponding quadratic approximation to the log-likelihood. The original model fit is included as the attribute "original.fit".

The plot method produces a plot of the likelihood slice for each parameter.

Author(s)

Rune Haubo B Christensen

Examples

## get data:
data(wine)
## fit model:


```r
fm1 <- clm(rating ~ contact + temp, data = wine)
## slice the likelihood:
sl1 <- slice(fm1)

## three different ways to plot the slices:
par(mfrow = c(2,3))
plot(sl1)
plot(sl1, type = "quadratic", plot.mle = FALSE)
plot(sl1, type = "linear")

## Verify convergence to the optimum:
sl2 <- slice(fm1, lambda = 1e-5, quad.approx = FALSE)
plot(sl2)
```

---

### Description

The `soup` data frame has 1847 rows and 13 variables. 185 respondents participated in an A-not A discrimination test with sureness. Before experimentation the respondents were familiarized with the reference product and during experimentation, the respondents were asked to rate samples on an ordered scale with six categories given by combinations of (reference, not reference) and (sure, not sure, guess) from 'referene, sure' = 1 to 'not reference, sure' = 6.

### Usage

`soup`

### Format

- **RESP** factor with 185 levels: the respondents in the study.
- **PROD** factor with 2 levels: index reference and test products.
- **PRODID** factor with 6 levels: index reference and the five test product variants.
- **SURENESS** ordered factor with 6 levels: the respondents ratings of soup samples.
- **DAY** factor with two levels: experimentation was split over two days.
- **SOUPTYPE** factor with three levels: the type of soup regularly consumed by the respondent.
- **SOUPFREQ** factor with 3 levels: the frequency with which the respondent consumes soup.
- **COLD** factor with two levels: does the respondent have a cold?
- **EASY** factor with ten levels: How easy did the respondent find the discrimination test? 1 = difficult, 10 = easy.
- **GENDER** factor with two levels: gender of the respondent.
- **AGEGROUP** factor with four levels: the age of the respondent.
- **LOCATION** factor with three levels: three different locations where experimentation took place.

### Source

Data are produced by Unilever Research. Permission to publish the data is granted.
References


update.clm2

**Update method for cumulative link models**

**Description**

Update method for cumulative link models fitted with `clm2`. This makes it possible to use e.g.

```r
update(obj, location = ~ . - var1, scale = ~ . + var2)
```

**Usage**

```r
## S3 method for class 'clm2'
update(object, formula., location, scale, nominal,...,
       evaluate = TRUE)
## S3 method for class 'clmm2'
update(object, formula., location, scale, nominal,...,
       evaluate = TRUE)
```

**Arguments**

- `object` a `clm2` object.
- `formula.` not used—unfortunately this argument is part of the default method.
- `location` an optional new formula for the location; see `update.formula` for details.
- `scale` an optional new formula for the scale; see `update.formula` for details.
- `nominal` an optional new formula for nominal effects; see `update.formula` for details.
- `...` additional arguments to the call, or arguments with changed values.
- `evaluate` if true evaluate the new call else return the call.

**Value**

If `evaluate = TRUE` the fitted object is returned, otherwise the updated call.

**Author(s)**

Rune Haubo B Christensen

**Examples**

```r
options(contrasts = c("contr.treatment", "contr.poly"))
data(soup)
m1 <- clm2(SURENESS ~ PROD, scale = ~PROD, data = soup,
           link = "logistic")
m2 <- update(m1, link = "probit")
m3 <- update(m1, link = "cloglog")
m4 <- update(m1, link = "loglog")
anova(m1, update(m1, scale = ~.-PROD))
```
## wine

### Description

The wine data set is adopted from Randall (1989) and from a factorial experiment on factors determining the bitterness of wine. Two treatment factors (temperature and contact) each have two levels. Temperature and contact between juice and skins can be controlled when crushing grapes during wine production. Nine judges each assessed wine from two bottles from each of the four treatment conditions, hence there are 72 observations in all.

### Usage

```r
wine
```

### Format

- **response**  scorings of wine bitterness on a 0—100 continuous scale.
- **rating**  ordered factor with 5 levels; a grouped version of response.
- **temp**  temperature: factor with two levels.
- **contact**  factor with two levels ("no" and "yes").
- **bottle**  factor with eight levels.
- **judge**  factor with nine levels.

### Source

Data are adopted from Randall (1989).

### References


Examples

data(wine)

## Variables 'rating' and 'response' are related in the following way:
(intervals <- seq(0,100, by = 20))
all(wine$rating == findInterval(wine$response, intervals)) ## ok

## A few illustrative tabulations:
## Table matching Table 5 in Randall (1989):
temp.contact.bottle <- with(wine, temp:contact:bottle)[drop=TRUE]
xtabs(response ~ temp.contact.bottle + judge, data = wine)

## Table matching Table 6 in Randall (1989):
with(wine, {
  tcb <- temp:contact:bottle
tcb <- tcb[drop=TRUE]
table(tcb, rating)
})
## or simply: with(wine, table(bottle, rating))

## Table matching Table 1 in Tutz & Hennevogl (1996):
tab <- xtabs(as.numeric(rating) ~ judge + temp.contact.bottle,
data = wine)
colnames(tab) <-
  paste(rep(c("c","w"), each = 4), rep(c("n","n","y","y"), 2),
  1:8, sep=".")
tab

## A simple model:
m1 <- clm(rating ~ temp * contact, data = wine)
summary(m1)
# Index

+ Topic **datasets**
- income, 37
- soup, 47
- wine, 49

+ Topic **distribution**
- gfun, 34
- gumbel, 35
- lgamma, 38

+ Topic **models**
- addterm.clm2, 4
- anova.clm2, 5
- clm, 7
- clm.control, 10
- clm.fit, 11
- clm2, 13
- clm2.control, 18
- clmm, 19
- clmm.control, 21
- clmm2, 22
- clmm2.control, 25
- confint, 26
- confint.clm2, 29
- convergence, 32
- drop.coef, 33
- predict.clm, 40
- predict.clm2, 42
- profile.clmm2, 43
- slice, 45
- update.clm, 48

+ Topic **package**
- ordinal-package, 2

<table>
<thead>
<tr>
<th>addterm, 6, 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>addterm.clm2, 4</td>
</tr>
<tr>
<td>addterm.default, 5</td>
</tr>
<tr>
<td>AIC, 8</td>
</tr>
<tr>
<td>anova, 5, 8, 14, 23</td>
</tr>
<tr>
<td>anova.clm2, 5</td>
</tr>
<tr>
<td>anova.clmm2 (anova.clm2), 5</td>
</tr>
<tr>
<td>anova.default, 6</td>
</tr>
</tbody>
</table>

| clm, 2, 3, 7, 11–13, 27, 32, 35, 36, 41 |
| clm.control, 7, 8, 10, 12, 20, 27 |
| clm.fit, 11 |
| clm2, 3–6, 8, 13, 18, 22, 23, 29, 30, 39, 42, 48 |
| clm2.control, 14, 18, 23 |
| clmm, 3, 19, 22, 35, 36, 41 |
| clmm.control, 19, 21 |
| clmm2, 3, 20, 22, 26, 39, 42, 44 |
| clmm2.control, 23, 25 |
| coeff, 8 |
| confint, 8, 14, 23, 26, 28, 31, 45 |
| confint.clm2, 29 |
| confint.clmm2 (profile.clmm2), 43 |
| confint.profile.clm2 (confint.clm2), 29 |
| confint.profile.clmm2 (profile.clmm2), 43 |
| convergence, 32 |
| data.frame, 28 |
| deprecated, 3 |
| dgumbel (gumbel), 35 |
| dlgamma (lgamma), 38 |
| drop.coef, 33 |
| drop1, 8 |
| dropterm, 6, 8, 14 |
| dropterm.clm2 (addterm.clm2), 4 |
| dropterm.default, 5 |
| extractAIC, 8 |
| formula, 7, 13 |
| gcauchy (gfun), 34 |
| gfun, 34, 36, 39 |
| ggumbel (gumbel), 35 |
| glgamma (lgamma), 38 |
| glm, 3 |
| glmer, 23 |
| glogis (gfun), 34 |
| gnorm (gfun), 34 |
| gumbel, 35, 35, 39 |
| income, 37 |
| lgamma, 36, 38 |
| lm, 33 |
| lmer, 3 |
log-gamma, 35
logLik, 14, 23
Matrix, 3
model.frame, 8, 9
model.matrix, 8, 33
napredict, 41
nlminb, 11, 18, 26
nobs, 8
optim, 11, 18, 26
ordinal (ordinal-package), 2
ordinal-package, 2
par, 27, 46
pgumbel (gumbel), 35
plgamma (lgamma), 38
plot.profile, 14, 23
plot.profile.clm (confint), 26
plot.profile.clm2 (confint.clm2), 29
plot.profile.clmm2 (profile.clmm2), 43
plot.slice.clm (slice), 45
polr, 15
predict, 14, 23, 24
predict.clm, 40
predict.clm2, 42
predict.clmm2 (predict.clm2), 42
predict.lm, 41
profile, 8, 14, 23, 28, 31, 45
profile.clm (confint), 26
profile.clm2 (confint.clm2), 29
profile.clmm2, 43
qgumbel (gumbel), 35
qr, 33
rgumbel (gumbel), 35
slice, 8, 45
soup, 47
step, 8
stepAIC, 8
summary, 8, 14, 23
ucminf, 11, 18, 21, 26
update, 14
update.clm2, 48
update.clmm2 (update.clm2), 48
update.formula, 48
vcov, 8, 14, 23
wine, 49
Appendix I

Reference manual for the R package sensR

Package ‘sensR’

February 6, 2012

Type Package

Title Thurstonian models for sensory discrimination

Version 1.2-16

Date 2012-02-06

Author Rune Haubo B Christensen and Per Bruun Brockhoff

Maintainer Rune Haubo B Christensen <rhbc@imm.dtu.dk>

Depends ordinal, numDeriv

Suggests MASS

Description Provides methods for sensory discrimination methods; duotrio, triangle, 2-AFC, 3-AFC, A-not A, same-different and 2-AC. This enables the calculation of d-primes, standard errors of d-primes, sample size and power computations, and comparisons of different d-primes. Methods for profile likelihood confidence intervals and plotting are included.

License GPL-2 | GPL-3

R topics documented:

AnotA ................................................................. 2
AUC ................................................................. 4
betabin .............................................................. 5
clls ................................................................. 8
clm2twoAC ......................................................... 10
confint.twoAC .................................................... 12
discrim ............................................................. 14
discrimPwr ......................................................... 16
discrimR ............................................................ 17
discrimSim ......................................................... 19
discrimSS .......................................................... 20
duotrio ............................................................... 22
finder ............................................................... 23
plot.discrim ....................................................... 24
plot.samediff .................................................... 25
AnotA

Description

Computation of dprime and its uncertainty for the monadic A-not-A test together with the one-tailed P-value of the difference test (Fisher's Exact test).

Usage

AnotA(x1, n1, x2, n2, 

## S3 method for class 'anota'
confint(object, parm, level = 0.95, 

## S3 method for class 'anota'
plot(x, main = TRUE, length = 1000, 

Arguments

x1 the number of (correct) A-answers on A-samples
n1 the total number of A-samples
x2 the number of A-answers on not-A-samples
n2 the number of not-A-samples
object an anota object
parm currently not used
level the desired confidence level
x an anota object
main should the plot have a main title?
length the discretization of the curves
... additional arguments passed to glm for AnotA; not used for confint and plot
Details

The AnotA function uses the glm and fisher.test functions of the stats package. Note that all arguments have to be positive integers.

Value

For AnotA an object of class anota (which has a print method). This is a list with elements

- coefficients: named vector of coefficients (d-prime)
- res.glm: the glm-object from the fitting process
- vcov: variance-covariance matrix of the coefficients
- se: named vector with standard error of the coefficients (standard error of d-prime)
- data: a named vector with the data supplied to the function
- p.value: one-sided p-value from Fisher’s exact test (fisher.test)
- test: a string with the name of the test (A-Not A) for the print method
- call: the matched call

For plot a figure of the distributions of sensory intensity is produced, and for confint a 2-by-2 matrix of confidence intervals is returned.

Author(s)

Rune Haubo B Christensen and Per Bruun Brockhoff

References


See Also

print.discrim, discrim, discrimPwr, discrimSim, discrimSS, findcr

Examples

# data: 10 of the A-samples were judged to be A
# 20 A-samples in total
# 3 of the not-A samples were judged to be A
# 20 not-A-samples in total
AnotA(10, 20, 3, 20)
(m1 <- AnotA(10, 20, 3, 20))

## plot distributions of sensory intensity:
plot(m1)

## likelihood based confidence intervals:
confint(m1)

## Extended example plotting the profile likelihood
xt <- cbind(c(3, 10), c(20 - 3, 20 - 10))
lev <- gl(2, 1)
summary(res <- glm(xt ~ lev,
AUC

**Description**

This is the default AUC function for scalar d-primes, which will compute Area Under the ROC curve (ROC is an acronym for receiver operating characteristic).

**Usage**

```r
## Default S3 method:
AUC(d, se.d, scale = 1, CI.alpha = 0.05, ...)

## S3 method for class 'anota'
AUC(d, CI.alpha = 0.05, ...)
```

**Arguments**

- **d** a unit length vector with the value of d-prime for which AUC is to be computed or a anota object from the fitting of a A-not A test with AnotA
- **scale** a unit length vector giving the ratio of scale (ie. standard deviation) of the latent distribution for the no-class items relative to that of the yes-class items
- **se.d** standard error of d (d-prime). If provided, the function will compute confidence limits of value of AUC—cf. in section value.
- **CI.alpha** the type I level of the confidence interval of AUC
- **...** additional arguments passed
betabin

Details

The function calls `integrate` to obtain the area under the ROC curve implied by `d` and `scale`. Confidence limits are based on a normal approximation of `d` and not of AUC. The limits are computed, if an estimate of the standard error of `d` is provided. Note that the limits do not take the uncertainty in estimating the scale nor that of estimating the standard error of `d` into account.

A print method is implemented for objects of class `AUC`.

Value

A list with components. If `se.d` is supplied to the default method or if a `discrim` object is supplied, the object contains the latter three additional elements.

- `value`: the estimated value of AUC
- `res.int`: the result from the call to `integrate`
- `lower`: the lower confidence limit
- `upper`: the upper confidence limit
- `CI.alpha`: echoes the provided `CI.alpha`

Author(s)

Rune Haubo B Christensen

Examples

```r
(odor <- matrix(c(112, 112, 72, 53, 22, 4, 7, 38, 5/zero.noslash, 117, 1/zero.noslash1, 62), 2,
byrow = TRUE))
(d.primes <- SDT(odor)[,3])
for(i in 1:5) print(AUC(d.primes[i]))
## Provide standard error of d-prime and compute CI:
fm1 <- AnotA(8, 25, 1, 25)
AUC(fm1$coef, fm1$se)
AUC(fm1)
```

betabin

*Beta-binomial and chance-corrected beta-binomial models for over-dispersed binomial data*

Description

Fits the beta-binomial model and the chance-corrected beta-binomial model to (over-dispersed) binomial data.

Usage

```r
betabin(data, start = c(.5,.5),
method = c("duotrio", "threeAFC", "twoAFC", "triangle"),
vcov = TRUE, corrected = TRUE, gradTol = 1e-4, ...)
## S3 method for class 'betabin'
summary(object, level = 0.95, ...)
```
Arguments

data matrix or data.frame with two columns; first column contains the number of
success and the second the total number of cases. The number of rows should
correspond to the number of observations.

start starting values to be used in the optimization

vcov logical, should the variance-covariance matrix of the parameters be computed?

method the sensory discrimination protocol for which d-prime and its standard error
should be computed

corrected should the chance corrected or the standard beta binomial model be estimated?

gradTol a warning is issued if max|gradient| < gradTol, where 'gradient' is the gradient
at the values at which the optimizer terminates. This is not used as a termination
or convergence criterion during model fitting.

object an object of class "betabin", i.e. the result of betabin().

level the confidence level of the confidence intervals computed by the summary method

... betabin: The only recognized (hidden) argument is doFit (boolean) which by
default is TRUE. When FALSE betabin returns an environment which facilitates
examination of the likelihood surface via the (hidden) functions sensR:::getParBB
and sensR:::setParBB. Not used in summary.betabin.

Details

The beta-binomial models are parameterized in terms of mu and gamma, where mu corresponds
to a probability parameter and gamma measures over-dispersion. Both parameters are restricted to
the interval (0, 1). The parameters of the standard (i.e. corrected = FALSE) beta-binomial model
refers to the mean (i.e. probability) and dispersion on the scale of the observations, i.e. on the scale
where we talk of a probability of a correct answer (Pc). The parameters of the chance corrected
(i.e. corrected = TRUE) beta-binomial model refers to the mean and dispersion on the scale of the
"probability of discrimination" (Pd). The mean parameter (mu) is therefore restricted to the interval
from zero to one in both models, but they have different interpretations.

The summary method use the estimate of mu to infer the parameters of the sensory experiment; Pc,
Pd and d-prime. These are restricted to their allowed ranges, e.g. Pc is always as least as large as
the guessing probability.

Confidens intervals are computed as Wald (normal-based) intervals on the mu-scale and the confi-
dence limits are subsequently transformed to the Pc, Pd and d-prime scales. Confidence limits are
restricted to the allowed ranges of the parameters, for example no confidence limits will be less than
zero.

Standard errors, and therefore also confidence intervals, are only available if the parameters are not
at the boundary of their allowed range (parameter space). If parameters are close to the boundaries
of their allowed range, standard errors, and also confidence intervals, may be misleading. The like-
lihood ratio tests are more accurate. More accurate confidence intervals such as profile likelihood
intervals may be implemented in the future.

The summary method provides a likelihood ratio test of over-dispersion on one degree of freedom
and a likelihood ratio test of association (i.e. where the null hypothesis is "no difference" and the
alternative hypothesis is "any difference") on two degrees of freedom (chi-square tests). Since the
gamma parameter is tested on the boundary of the parameter space, the correct degree of freedom
for the first test is probably 1/2 rather than one, or somewhere in between, and the latter test is
probably also on less than two degrees of freedom. Research is needed to determine the appropriate
no. degrees of freedom to use in each case. The choices used here are believed to be conservative,
so the stated p-values are probably a little too large.
The log-likelihood of the standard beta-binomial model is
\[
\ell(\alpha, \beta; x, n) = -N \log \text{Beta}(\alpha, \beta) + \sum_{j=1}^{N} \log \text{Beta}(\alpha + x_j, \beta - x_j + n_j)
\]

and the log-likelihood of the chance corrected beta-binomial model is
\[
\ell(\alpha, \beta; x, n) = -N \log \text{Beta}(\alpha, \beta) + \sum_{j=1}^{N} \log \left\{ \frac{x_j}{\sum_{i=1}^{n} (x_j \cdot p_{ij})} \right\} (1 - p_{ij})^{n_j - x_j + 1} \text{Beta}(\alpha + i, n_j - x_j + \beta)
\]

where \( \mu = \alpha / (\alpha + \beta) \), \( \gamma = 1 / (\alpha + \beta + 1) \), \( \text{Beta} \) is the Beta function, cf. \( \text{beta} \), \( N \) is the number of independent binomial observations, i.e. the number of rows in \( \text{data} \), and \( p_g \) is the guessing probability, \( p_{\text{Guess}} \).

The variance-covariance matrix (and standard errors) is based on the inverted Hessian at the optimum. The Hessian is obtained with the \( \text{hessian} \) function from the \text{numDeriv} package.

The gradient at the optimum is evaluated with \( \text{gradient} \) from the \text{numDeriv} package.

The bounded optimization is performed with the "L-BFGS-B" optimizer in \text{optim}.

The following additional methods are implemented objects of class \text{betabin}: \text{print}, \text{vcov} and \text{logLik}.

\textbf{Value}

An object of class \text{betabin} with elements

- \text{coefficients} named vector of coefficients
- \text{vcov} variance-covariance matrix of the parameter estimates if \text{vcov} = \text{TRUE}
- \text{data} the data supplied to the function
- \text{call} the matched call
- \text{logLik} the value of the log-likelihood at the MLEs
- \text{method} the method used for the fit
- \text{convergence} 0 indicates convergence. For other error messages, see \text{optim}.
- \text{message} possible error message - see \text{optim} for details
- \text{counts} the number of iterations used in the optimization - see \text{optim} for details
- \text{corrected} is the chance corrected model estimated?
- \text{logLikNull} log-likelihood of the binomial model with \text{prop} = p_{\text{Guess}}
- \text{logLikMu} log-likelihood of a binomial model with \text{prop} = \text{sum(x)/sum(n)}

\textbf{Author(s)}

Rune Haubo B Christensen

\textbf{References}


\textbf{See Also}

\text{triangle}, \text{twoAFC}, \text{threeAFC}, \text{duotrio},
## Examples

```r
defines x and n as vectors
x <- c(3,2,6,8,3,4,6,9,9,2,1,2,8,9,5,7)
n <- c(10,9,8,9,8,6,9,10,10,10,10,10,10,10,9,10)

# Create data:
dat <- data.frame(x, n)

bb <- betabin(dat, method = "duotrio")
bb <- betabin(dat, corrected = FALSE, method = "duotrio")
summary(bb)
vcov(bb)
logLik(bb)
AIC(bb)
coef(bb)
```

---

### clls

**Cumulative Link Location-Scale Models**

**Description**

**IMPORTANT:** This function and its methods are no longer supported. The user is advised to use `clm()` from package ordinal instead.

Fits a cumulative link location-scale model to an ordered response variable. When the scale part is left unspecified, the model reduces to a cumulative link model assuming a constant scale. With the default logistic link function, the model reduces to the famous *Proportional Odds Model*. With the probit link and a single two-level factor in both location and scale parts, the model is known as the *Binormal* model in the Signal Detection Theory and the Psychometric literature.

**Usage**

```r
clls(location, scale, data, weights, start, ..., subset, na.action, contrasts = NULL, Hess = FALSE, model = TRUE, method = c("logistic", "probit", "cloglog", "cauchit"))
```

**Arguments**

- **location**: a formula expression as for regression models, of the form `response ~ predictors`. The response should be a factor (preferably an ordered factor), which will be interpreted as an ordinal response, with levels ordered as in the factor. The model must have an intercept: attempts to remove one will lead to a warning and be ignored. An offset may be used. See the documentation of `formula` for other details.

- **scale**: a optional formula expression as for the location part, of the form `~ predictors`, i.e. with an empty left hand side. If left unspecified, the model assumes a constant scale and reduces to the cumulative link model. An offset may be used. See the documentation of `formula` for other details.

- **data**: an optional data frame in which to interpret the variables occurring in `formula`.

- **weights**: optional case weights in fitting. Default to 1.

- **start**: initial values for the parameters. This is in the format `c(beta, theta, sigma)`: see the Values section.
... additional arguments to be passed to `optim`, most often a `control` argument.

`subset` expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.

`na.action` a function to filter missing data.

`contrasts` a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.

`Hess` logical for whether the Hessian (the observed information matrix) should be returned. Use this if you intend to call `summary` or `vcov` on the fit.

`model` logical for whether the model matrix should be returned.

`method` logistic or probit or complementary log-log or cauchit (corresponding to a Cauchy latent variable).

**Details**

The implementation is highly inspired by `polr` in package MASS and should give compatible results, if `scale` is left unspecified.

Note that standard errors are appropriate for `tau = \log \sigma` and not for `sigma`, because the profile likelihood is usually more symmetric for `tau` than for `sigma`. Therefore `vcov` will give the variance-covariance matrix of the parameters with `tau` rather than `sigma` and `summary.clls` will report standard errors for `log sigma`. Notice also that a relevant test for `sigma` is $H_0 : sigma = 1$, so the relevant test for `log sigma` is $H_0 : log(sigma) = 0$. This is reflected in the z value for `sigma` returned by `summary.clls`.

There are methods for the standard model-fitting functions, including `summary`, `vcov`, `anova`, and an `extractAIC` method.

**Value**

A object of class "clls". This has components

- `coefficients` the coefficients of the location (beta), the intercepts (theta) and the scale (sigma).
- `beta` the parameter estimates of the location part.
- `theta` the intercepts/thresholds for the class boundaries.
- `sigma` the parameter estimates of the scale part.
- `tau` parameter estimates of the scale part on the log scale; i.e. \( \tau = \log \sigma \).
- `deviance` the residual deviance.
- `fitted.values` a matrix, with a column for each level of the response with the fitted probabilities.
- `fitted.case` a vector of same length as `response`, with the fitted probabilities on a case-by-case basis.
- `lev` the names of the response levels.
- `terms.location` a terms structure describing the location part.
- `terms.scale` a terms structure describing the scale part.
- `df.residual` the number of residual degrees of freedoms, calculated using the weights.
- `edf` the (effective) number of degrees of freedom used by the model.
- `n`, `nobs` the (effective) number of observations, calculated using the weights.
call  the matched call.
method  the matched method used.
convergence  the convergence code returned by optim.
niter  the number of function and gradient evaluations used by optim.
Hessian  if Hess is true, the observed Fisher information matrix.
location  if model is true, the model.frame for the location part.
scale  if model is true, the model.frame for the scale part.

Reference


See Also

polr, optim, glm, multinom.

Examples

```r
options(contrasts = c("contr.treatment", "contr.poly"))
## Extend example from polr in package MASS:
## Fit model from polr example:
data(housing, package = "MASS")
fm1 <- clns(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
fm1
summary(fm1)
## With probit link:
summary(update(fm1, method = "probit"))

## Allow scale to depend on Cont-variable
summary(fm2 <- update(fm1, scale =~ Cont))
anova(fm1, fm2)
## which seems to improve the fit
```

---

clm2twoAC  

**Extract 2-AC coefficient table from a cumulative link model**

**Description**

The Thurstonian model for the 2-AC protocol can be formulated as a cumulative link model (see the references). This function extracts the 2-AC model parameter estimates, standard errors, z-value and p-values from a cumulative link (mixed) model fitted with clm or clmm from package ordinal.

**Usage**

```r
clm2twoAC(object, ...)
```
Arguments

object a `clm` or `clmm` object
...

Value

A `data.frame` with the coefficient table. The two first rows contain the estimates of `tau` and `d.prime` while the remaining rows contain optional regression variables for `d.prime`.

Author(s)

Rune Haubo B Christensen

References


See Also

twoAC, twoACpwr

Examples

```r
## Example of a simple 2-AC model. First the conventional way:
twoAC(c(2, 2, 6))
## The using a cumulative link model (clm from package ordinal):
library(ordinal)
response <- gl(3,1)
fit.clm <- clm(response ~ 1, weights = c(2, 2, 6), link = "probit")
clm2twoAC(fit.clm)
## Alternatively we could get estimates and standard errors "by hand":
tab <- coef(summary(fit.clm))
theta <- tab[,1]
(tau <- (theta[2] - theta[1])/sqrt(2))
(d.prime <- (-theta[2] - theta[1])/sqrt(2))
VCOV <- vcov(fit.clm)
(se.tau <- sqrt((VCOV[1,1] + VCOV[2,2] - 2*VCOV[2,1])/2))
(se.d.prime <- sqrt((VCOV[1,1] + VCOV[2,2] + 2*VCOV[2,1])/2))
## Extended example with a regression model for d.prime
## (see the referenced paper for details):
n.women <- c(2, 2, 6)*1/zero.noslash
n.men <- c(1, 2, 7)*1/zero.noslash
wt <- c(n.women, n.men)
response <- gl(3,1, length = 6)
gender <- gl(2, 3, labels = c("women", "men"))
fm2 <- clm(response ~ gender, weights = wt, link = "probit")
clm2twoAC(fm2)
```
### Description

Computes confidence intervals from the profiled likelihood and the Wald approximation in the 2AC model, or plots the profile likelihood function for d.prime.

### Usage

```r
## S3 method for class 'twoAC'
confint(object, parm, level = 0.95,
        type = c("likelihood", "Wald"), ...)

## S3 method for class 'profile.twoAC'
confint(object, parm = "d.prime", level = 0.95, ...)

## S3 method for class 'twoAC'
profile(fitted, alpha = 1e-3, nSteps = 1e2, range, ...)

## S3 method for class 'profile.twoAC'
plot(x, level = c(0.95, 0.99), Log = FALSE,
     relative = TRUE, fig = TRUE, n = 1e3, ..., ylim = NULL)
```

### Arguments

- `object`: A fitted `twoAC` object or a `profile.twoAC` object.
- `fitted`: A fitted `twoAC` object.
- `x`: A `profile.twoAC` object.
- `type`: The type of confidence interval required. "profile" is the most accurate.
- `parm`: For `confint.profile.twoAC`: has to be "d.prime".
  For `confint.twoAC`: for `type = "Wald"` a specification of which parameters the confidence interval is required for. Ignored for `type = "profile"`.
- `level`: The confidence level required.
- `alpha`: Determines the range of profiling. By default the likelihood is profiled in the 99.9% Wald confidence interval region.
- `range`: If supplied, d.prime will be profiled between `min(range)` and `max(range)`. This over-rules the automatic range computation.
- `nSteps`: The number of profile steps.
- `Log`: Should the profile likelihood be plotted on the log-scale?
- `relative`: Should the relative or the absolute likelihood be plotted?
- `fig`: Should the profile likelihood be plotted?
- `n`: The no. points used in the spline interpolation of the profile likelihood.
- `ylim`: Overrules default y-limits on the plot of the profile likelihood.
- `...`: Not currently used.
Details

These `confint` methods call the appropriate profile method, then finds the confidence intervals by interpolation of the profile traces. If the profile object is already available, this should be used as the main argument rather than the fitted model object itself.

In `plot.profile.twoAC`: at least one of `Log` and `relative` arguments have to be `TRUE`.

Value

`confint`: A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as `(1-level)/2` and `1 - (1-level)/2` in % (by default 2.5% and 97.5%). Profile likelihood confidence intervals are only available for `d.prime` and not `tau`.

`profile.twoAC`: a `data.frame` with the profile of `d.prime`.

`plot.profile.twoAC` invisibly returns the spline approximation to the profile.

Author(s)

Rune Haubo B Christensen

References


See Also

`profile` and `confint`

Examples

```r
(fm1 <- twoAC(c(2, 2, 6)))
confint(fm1)
confint(fm1, type = "Wald")

pr1 <- profile(fm1)
confint(pr1)

pr1 <- profile(fm1, alpha = 1e-5)
par(mfrow = c(2,2))
plot(pr1)
plot(pr1, Log = FALSE, relative = TRUE)
plot(pr1, Log = TRUE, relative = TRUE)
plot(pr1, Log = TRUE, relative = FALSE)
```
Sensory discrimination analysis

Description
Computes the probability of a correct answer (Pc), the probability of discrimination (Pd) and d-prime, their standard errors, confidence intervals and a p-value of a difference or similarity test for one of the four common discrimination protocols.

Usage

discrim(correct, total, pd0 = 0, conf.level = 0.95, 
method = c("duotrio", "threeAFC", "twoAFC", "triangle"), 
statistic = c("exact", "likelihood", "score", "Wald"), 
test = c("difference", "similarity"), ...)

## S3 method for class 'discrim'
print(x, digits = getOption("digits"), ...)

Arguments

- **correct**: the number of correct answers; non-negative scalar integer
- **total**: the total number of answers (the sample size); positive scalar integer
- **pd0**: the probability of discrimination under the null hypothesis; numerical scalar between zero and one
- **conf.level**: the confidence level for the confidence intervals
- **method**: the discrimination protocol. Four allowed values: "twoAFC", "threeAFC", "duotrio", "triangle"
- **test**: the type of test
- **statistic**: the statistic to be used for hypothesis testing and confidence intervals
- **x**: an object of class "discrim"
- **digits**: number of digits in resulting table of results
- **...**: not currently used

Details
The p-value for the standard one-tailed difference test of "no difference" is obtained with pd0 = 0. The probability under the null hypothesis is given by pd0 + pg * (1 - pd0) where pg is the guessing probability which is defined by the discrimination protocol given in the method argument.

All estimates are restricted to their allowed ranges, e.g. Pc is always as least as large as the guessing probability. Similarly confidence limits are also restricted to the allowed range of the parameters.

Standard errors are not defined when the parameter estimates are at the boundary of their allowed range, so these will be reported as NA in such cases.
The "Wald" statistic is *NOT* recommended for practical use—it is included here for completeness and to allow comparisons.

For statistic = "score", the confidence interval is computed from Wilson's score interval, while the p-value for the hypothesis test is based on Pearson's chi-square test, cf. \texttt{prop.test}.

**Value**

An object of class \texttt{discrim} with elements

- \texttt{coefficients}: matrix of estimates, standard errors and confidence intervals
- \texttt{data}: a named vector with the data supplied to the function
- \texttt{p.value}: the p-value of the hypothesis test
- \texttt{call}: the matched call
- \texttt{test}: the type of test
- \texttt{method}: the discrimination protocol
- \texttt{statistic}: the statistic used for confidence intervals and p-value
- \texttt{pd0}: the probability of discrimination under the null hypothesis
- \texttt{conf.level}: the confidence level
- \texttt{stat.value} for statistic \(!= "exact\)" the value of the test statistic used to calculate the p-value
- \texttt{df} for statistic \(!= "score\)" the number of degrees of freedom used for the Pearson chi-square test to calculate the p-value
- \texttt{profile} for statistic \(!= "likelihood\)" the profile likelihood on the scale of \(P_c\)

**Author(s)**

Rune Haubo B Christensen and Per Bruun Brockhoff

**References**


**See Also**

\texttt{triangle, twoAFC, threeAFC, duotrio, discrimPwr, discrimSim, discrimSS, samediff, AnotA, findcr, profile, plot.profile confint}

**Examples**

```r
## Running the simple discrimination (difference) tests:
discrim(10, 15, method = "twoAFC")
discrim(10, 15, method = "threeAFC", statistic = "likelihood")
discrim(10, 15, method = "duotrio", conf.level = 0.90)
discrim(10, 15, method = "triangle", statistic = "score")
```

```r
## plot the distributions of sensory intensity:
m1 <- discrim(10, 15, method = "twoAFC")
plot(m1)
```

```r
## A similarity test where less than chance successes are obtained:
```
discrimPwr

Sensory discrimination power analysis

Description
Computes the power of a difference or similarity test for a sensory discrimination experiment using the binomial distribution. d.primePwr is a convenience function that calls discrimPwr but has arguments in terms of d-prime rather than pd, the probability of discrimination.

Usage

discrimPwr(pdA, pd0 = 0, sample.size, alpha = 0.05, pGuess = 1/2,
  test = c("difference", "similarity"),
  statistic = c("exact", "normal"))
d.primePwr(d.primeA, d.prime0 = 0, sample.size, alpha = 0.05,
  method = c("duotrio", "threeAFC", "twoAFC", "triangle"),
  test = c("difference", "similarity"),
  statistic = c("exact", "normal"))

Arguments

- **pdA**: the probability of discrimination for the model under the alternative hypothesis; scalar between zero and one
- **d.primeA**: d-prime for the model under the alternative hypothesis; non-negative numerical scalar
- **pd0**: the probability of discrimination under the null hypothesis; scalar between zero and one
- **d.prime0**: d-prime under the null hypothesis; non-negative numerical scalar
- **sample.size**: the sample size; a scalar positive integer
- **alpha**: the type I level of the test; scalar between zero and one
- **method**: the discrimination protocol for which the power should be computed
- **pGuess**: the guessing probability for the discrimination protocol, e.g. 1/2 for duo-trio and 2-AFC; and 1/3 for triangle and 3-AFC; scalar between zero and one
- **test**: the type of one-sided binomial test (direction of the alternative hypothesis): "difference" corresponds "greater" and "similarity" corresponds to "less"
- **statistic**: should power determination be based on the 'exact' binomial test or the normal approximation to this?

Details
The power of the standard one-tailed difference test where the null hypothesis is "no difference" is obtained with pd0 = 0.

The probability under the null hypothesis is given by pd0 + pg * (1 - pd0) where pg is the guessing probability pGuess. Similarly, the probability of the alternative hypothesis is given by pdA + pg * (1 - pdA)
Value

The power; a numerical scalar.

Author(s)

Rune Haubo B Christensen and Per Bruun Brockhoff

References


See Also

findcr, discrim, discrimSim, AnotA, discrimSS

Examples

## Finding the power of a discrimination test with d-prime = 1, ## a sample of size 30 and a type I level of .05:
pd <- coef(rescale(d.prime = 1, method = "twoAFC"))$pd
discrimPwr(pd, sample.size = 30)
discrimPwr(1, sample.size = 30, method = "twoAFC")

## A similarity example:
discrimPwr(pdA = .1, pd0 = .2, sample.size = 100, pGuess = 1/3,
test = "similarity")

discrimR

Replicated Thurstonian Model for discrimination analysis

Description

The model is a synthesis of a mixture and a mixed effect model. The random effect distribution for the cluster term (often individuals) is a point mass for delta = 0 and a continuous distribution for delta > 0.

The function fits the model and computes d-prime for an average subject, 2) the variance among subjects, 3) the "posterior" probability of a subject being a discriminator (with delta > 0), 4) the "posterior" expectation on the random effect (ie. the subject-specific delta) and 5) the probability that a randomly chosen individual is a discriminator (ie. the probability mass at delta = 0 in the random effects distribution)

Warning: This function is preliminary; see the details for further information.

Usage

discrimR(formula, data, weights, cluster, start, subset, na.action, contrasts = NULL, hess = FALSE, ranef = FALSE, zi = FALSE, method = c("duotrio", "probit", "threeAFC", "triangle", "twoAFC"), ...)
Arguments

**formula**  
A formula where the lhs is the binomial response. An indicator vector or a matrix with two column; successes and failures like in a call to `glm` with a binomial family. The rhs should be 1; no other predictors are currently allowed, but extending this is ongoing work.

**data**  
The data.frame in which to look for variables.

**weights**  
Possible weights

**cluster**  
The clustering variable; should be a factor.

**start**  
Optional starting values; recommended in the current implementation

**subset**  
...

**na.action**  
...

**contrasts**  
...

**hess**  
Should the hessian of the parameters be computed?

**ranef**  
Should the random effect estimates be computed?

**zi**  
Should the posterior probabilities of a subject being a discriminator be computed?

**method**  
Should correspond to the actual test applied.

**...**  
Additional arguments to `optim`. control=list(trace=TRUE, REPORT=1) is recommended, so the reduction in deviance and convergence can be followed.

Details

This function is preliminary and improving it is ongoing work. The computational methods are expected to change completely. This will hopefully facilitate methods for more general rhs-formulae with additional predictors.

Currently no methods or extractor functions have been written, so the user will have to select the relevant elements from the fitted object (see below). Implementation of methods and extractor functions will occur in due course.

Value

A list with the following elements:

**fpar**  
The fixed effect parameter, ie. delta (for an average individual)

**rpar**  
A vector with two elements: The first element is the variance component (standard deviation) on the log-scale, where optimization is performed. The second element is the variance component (standard deviation) on the original scale.

**deviance**  
Deviance for the model

**se**  
standard errors for 1) the fixed effect parameter and 2) the variance component on the log-scale

**convergence**  
Convergence message from `optim`

**lli**  
Log-likelihood contributions from each of the observations.

**ranef**  
The random effect estimates for the levels of the clustering factor (often individual)

**zi**  
posterior probabilities of a subject being a discriminator

**p**  
The probability that a randomly chosen individual is a discriminator (ie. the probability mass for delta > 0 in the random effects distribution)
**discrimSim**

Simulates replicated difference tests

**Description**

Simulates the outcome of sample.size replicated sensory difference tests (for any one of four protocols: 2-AFC, 3-AFC, duotrio and triangle tests) for a given d-prime value and a given overdispersion (default 0).

**Usage**

```r
discrimSim(sample.size, replicates, d.prime, sd.indiv = 0, method = c("duotrio", "halfprobit", "probit", "triangle", "twoAFC", "threeAFC"))
```

**Arguments**

- `sample.size`: the sample size - number of subjects
- `replicates`: number of replications per subject
- `d.prime`: the value of d-prime
- `method`: the discrimination protocol

**Examples**

```r
## Not run:
freq <- c(10, 8, 10, 9, 8, 9, 9, 1, 10, 8, 2, 6, 7, 6, 4, 5, 5, 3, 9, 9, 5, 5, 8, 8, 9, 9)
tmp <- data.frame(id = factor(1:3), n = rep(10, 3), freq = freq)
head(tmp)
str(tmp)

fm <- discrimR(cbind(freq, n - freq) ~ 1, tmp, cluster = id,
               start = c(.5, .5), method = "twoAFC",
               ranef = TRUE, zi = TRUE, hess = TRUE,
               control=list(trace=TRUE, REPORT=1))

names(fm)
fm[1:4]

## End(Not run)
```
sd.indiv  the individual variability in d-prime values. A value of 0 (default) corresponds
to complete independence

Details
The d-prime for each subject is a random draw from a normal distribution with mean d.prime and
standard deviation sd.indiv. All negative values are set to zero.

Value
A vector of length sample.size with the number of correct answers for each subject.

Author(s)
Rune Haubo B Christensen and Per Bruun Brockhoff

References

See Also
triangle, twoAFC, threeAFC, duotrio, discrimPwr, discrim, AnotA, discrimSS, samediff, findcr

Examples

```r
## Running simulations:
discrimSim(sample.size = 10, replicates = 3, d.prime = 2,
            method = "triangle", sd.indiv = 1)
```

### Value
A vector of length sample.size with the number of correct answers for each subject.

Author(s)
Rune Haubo B Christensen and Per Bruun Brockhoff

References

See Also
triangle, twoAFC, threeAFC, duotrio, discrimPwr, discrim, AnotA, discrimSS, samediff, findcr

Examples

```r
## Running simulations:
discrimSim(sample.size = 10, replicates = 3, d.prime = 2,
            method = "triangle", sd.indiv = 1)
```

### Description
Computes the sample size for a difference or similarity test for a sensory discrimination experiment
using the binomial distribution. The function returns the smallest sample size for the test that gives
the at least the desired target-power for the one-sided exact binomial test. A normal approximation
to the sample size / power may used. d.primeSS is a convenience function that calls discrimSS
but has arguments in terms of d-prime rather than pd, the probability of discrimination.

Usage

```r
discrimSS(pdA, pd0 = 0, target.power = 0.90, alpha = 0.05,
          pGuess = 1/2, test = c("difference", "similarity"),
          statistic = c("exact", "normal"))
dprimeSS(d.primeA, d.prime0 = 0, target.power = 0.90, alpha = 0.05,
        method = c("duotrio", "threeAFC", "twoAFC", "triangle"),
        test = c("difference", "similarity"),
        statistic = c("exact", "normal"))
```
Arguments

- **pdA**: the probability of discrimination for the model under the alternative hypothesis; scalar between zero and one.
- **d.primeA**: d-prime for the model under the alternative hypothesis; non-negative numerical scalar.
- **pd0**: the probability of discrimination under the null hypothesis; scalar between zero and one.
- **d.prime0**: d-prime under the null hypothesis; non-negative numerical scalar.
- **target.power**: the desired power for the test.
- **alpha**: the type I level of the test; scalar between zero and one.
- **method**: the discrimination protocol for which the sample size should be computed.
- **pGuess**: the guessing probability for the discrimination protocol, e.g. 1/2 for duo-trio and 2-AFC, and 1/3 for triangle and 3-AFC; scalar between zero and one.
- **test**: the type of one-sided binomial test (direction of the alternative hypothesis): "difference" corresponds "greater" and "similarity" corresponds to "less".
- **statistic**: should sample size calculation be based on the 'exact' binomial test or the normal approximation to this?

Details

The sample size of the standard one-tailed difference test where the null hypothesis of "no difference" is obtained with pd0 = 0.

The probability under the null hypothesis is given by pd0 + pg * (1 - pd0) where pg is the guessing probability pGuess. Similarly, the probability of the alternative hypothesis is given by pdA + pg * (1 - pdA).

The sample size for the 'exact' test, the function uses the normal approximation to get a starting value for an iterative search. If the sample size based on the normal approximation is larger than 10,000, the function returns the normal approximation and issues a warning.

Value

The sample size; a scalar integer.

Author(s)

Per Bruun Brockhoff and Rune Haubo B Christensen

References


See Also

AnotA, discrimPwr, sameDiff, findcr
Examples

```r
## Finding the smallest necessary sample size:
   discrimSS(pdA = 0.5, pd0 = 0, target.power = 0.80, alpha = 0.05,
             pGuess = 1/2, test = "difference", statistic = "exact")
```

```r
## Give identical results:
   pd <- coef(rescale(d.prime = 1, method = "twoAFC"))$pd
   discrimSS(pdA = pd, pd0 = 0, target.power = 0.90, alpha = 0.05,
             pGuess = 1/2, test = "difference", statistic = "exact")
   d.primeSS(1, target.power = 0.90, method = "twoAFC")
```

```r
## A similarity example:
   discrimSS(pdA = 0.1, pd0 = 0.2, target.power = 0.80, alpha = 0.05,
             pGuess = 1/2, test = "similarity", statistic = "exact")
```

---

duotrio

Create duotrio binomial family

Description

Creates a copy of the binomial family with the inverse link function changed to equal the duotrio psychometric function and correspondingly changed link function and derivative of the inverse link function.

Usage

```r
duotrio()
```

Value

A binomial family object for models. Among other things it includes the psychometric function as `linkinv` and the inverse psychometric function (for direct dprime computation) as `linkfun`.

Note

Several functions in this package make use of the function, but it may also be used on its own—see the example below.

Author(s)

Per Bruun Brockhoff

References


See Also

`triangle`, `twoAFC`, `threeAFC`, `discrim`, `discrimPwr`, `discrimSim`, `AnotA`, `discrimSS`, `samediff`, `findcr`
Examples

xt <- matrix(c(1/zero.noslash, 5), ncol = 2) ## data: 1/zero.noslash correct answers, 5 incorrect
res <- glm(xt ~ 1, family = duotrio)
summary(res)

## Extended example plotting the profile likelihood
## data: 1/zero.noslash correct answers, 5 incorrect
xt <- matrix(c(1/zero.noslash, 5), ncol = 2)
summary(res <- glm(xt ~ 1, family = duotrio))
N <- 100
dev <- double(N)
delta <- seq(1e-4, 5, length = N)
for(i in 1:N)
  dev[i] <- glm(xt ~ -1 + offset(delta[i]),
    family = duotrio)$deviance
plot(delta, exp(-dev/2), type = "l",
    xlab = expression(delta),
    ylab = "Normalized Profile Likelihood")
## Add Normal approximation:
lines(delta, exp(-(delta - coef(res))^2 /
    (2 * vcov(res))), lty = 2)
## Add confidence limits:
level <- c(0.95, 0.99)
lim <- sapply(level, function(x)
    exp(-qchisq(x, df=1)/2) )
abline(h = lim, col = "grey")
points(confint(res), rep(lim[1], 2), pch = 4)

findcr

Find the critical value of a one-tailed binomial test

Description

Finds the critical value in a one-tailed binomial test

Usage

findcr(sample.size, alpha = .05, p0 = .5, pd0 = 0,
    test = c("difference", "similarity"))

Arguments

sample.size the sample size of the binomial test (must be a positive integer)
alpha the test I error-level of the test (must be between zero and one)
p0 the guessing probability under the null-hypothesis (must be between zero and one): 1/2 for the duotrio and twoAFC tests and 1/3 for the triangle and threeAFC tests
pd0 the proportion of discriminators in the population of interest
test the type of test
plot.discrim

Details

The critical value of the standard one-tailed difference test of "no difference" is obtained with \( pd_0 = 0 \).

The probability of a correct answer under the null hypothesis is given by \( pd_0 + p_0 \times (1 - pd_0) \).

Value

The critical value in a one-tailed binomial test, that is, the smallest integer such that the null hypothesis binomial probability of being larger (smaller for similarity hypotheses) than or equal to this number is smaller than or equal to the type I error-level of the test.

Author(s)

Rune Haubo B Christensen and Per Bruun Brockhoff

See Also

triangle, twoAFC, threeAFC, duotrio, discrim, discrimPwr, discrimSim, AnotA discrimSS, samediff

Examples

## Find the critical value for a triangle test for the level 0.05 test
## with 25 subjects:
findcr(sample.size = 25, , p0 = 1/3)

## Similarity example:
findcr(sample.size = 25, p0 = 1/3, pd0 = .2, test = "simil")

plot.discrim

Plot function for discrim objects

Description

This function plots the latent distributions of sensory intensity corresponding to the items or products tested in the discrimination test.

Usage

## S3 method for class 'discrim'
plot(x, main = TRUE, length = 1000, ...)

Arguments

x

The discrim object whose latent distributions are to be plotted

main

include an automatically generated title on the plot? Default is TRUE

length

the length of the vectors to be plotted. Longer vectors gives more smooth curves.

...

additional arguments to plot and lines

Author(s)

Rune Haubo B Christensen
Examples

```r
## Generate discrim objects to be plotted:
fm1 <- discrim(10, 15, method = "threeAFC")
fm2 <- discrim(10, 15, method = "triangle")
par(mfrow=c(2,1)) ## Split plotting window in two
## Plot the distributions of sensory intensity for the two objects
## and increase the line width
plot(fm1, lwd=2)
plot(fm2, lwd=2)
```

Description

This function plots the latent distributions of sensory intensity corresponding to the items or products tested in the discrimination test.

Usage

```r
## S3 method for class 'samediff'
plot(x, main = TRUE, length = 1/zero.noslash/zero.noslash/zero.noslash,
     limits, fig = TRUE, ...)
```

Arguments

- `x`: The `samediff` object whose latent distributions are to be plotted
- `main`: include an automatically generated title on the plot? Default is `TRUE`
- `length`: the length of the vectors to be plotted. Longer vectors gives more smooth curves, but can take a little time.
- `limits`: optional limits on the x-axis; vector of length two.
- `fig`: logical: Should the function create the plot? Defaults to `TRUE`.
- `...`: additional arguments to `plot` and `lines`

Value

If `fig = TRUE`, the function will produce the plot. The function invisibly returns a data.frame with elements

- `z`: values for the x-axis of length `length`.
- `base.dist`: y-values for the base distribution of same-samples, i.e. a standard normal distribution
- `delta.dist`: y-values for the distribution of different-samples, i.e. a normal distribution centred at `delta` with unit variance.

This facilitates later plotting and changing the appearance of the plot.

Author(s)

Rune Haubo B Christensen
Examples

```r
## Make same-diff object:
sadi <- samediff(8, 5, 4, 9)
## Plot distributions of sensory intensity:
plot(sadi)
```

Description

Computes the (normalized or relative) profile likelihood for the parameters of a discrimination test, plots the normalized profile likelihood.

Usage

```r
## S3 method for class 'discrim'
profile(fitted, ...)

## S3 method for class 'profile.discrim'
plot(x, level = c(0.99, 0.95), fig = TRUE, method = "natural", n = 1e3, ...)

## S3 method for class 'discrim'
confint(object, parm, level = 0.95, ...)
```

Arguments

- `fitted`: a `discrim` object
- `x`: a `profile.discrim` object
- `object`: a `discrim` object
- `parm`: currently not used
- `method`: the type of spline to be used in approximating the profile likelihood curve (trace)—see `spline` for details
- `n`: the number of spline interpolations to use in plotting the profile likelihood curve (trace)
- `level`: for `plot`: At which levels to include horizontal lines to indicate confidence levels in plots of the normalized profile likelihoods. For `confint`: at which level to compute the confidence interval
- `fig`: logical: should the normalized profile likelihoods be plotted?
- `...`: For `plot`: additional arguments to `plot`. Otherwise not used.

Details

`confint` returns the confidence interval computed in `discrim` possibly at another level. The statistic used to compute the confidence interval is therefore determined in the `discrim` call and may not be the likelihood root.

The likelihood profile is extracted from the `discrim` object fitted with `statistic = "likelihood"`. 

profile.discrim

Profile likelihood and confidence interval methods for discrim objects
Value

For profile: An object of class "profile.discrim", "data.frame"—a data.frame with two columns giving the value of the parameter and the corresponding value of the profile likelihood.

For plot: The profile object is returned invisibly.

For confint:
A 3x2 matrix with columns named "lower", "upper" giving the lower and upper (100 * level)% confidence interval for the parameters named in the rows.

Author(s)

Rune Haubo B Christensen and Per Bruun Brockhoff

References


See Also
discrim

Examples

## 7 success out of 10 samples in a duo-trio experiment:
(dd <- discrim(7, 10, method = "duotrio", statistic = "likelihood"))
confint(dd)
plot(profile(dd))
points(confint(dd)[3,], rep(.1465, 2), pch = 3, cex = 2, lwd=2)

profile.samediff  Profile likelihood methods for samediff objects.

Description

Computes the (normalized or relative) profile likelihood for the parameters of a same-different test, plots the normalized profile likelihood and computes profile likelihood confidence intervals.

Usage

## S3 method for class 'samediff'
profile(fitted, which = 1:2, max = 2, numpts = 1/zero.noslash/zero.noslash,
        max.delta = 1/zero.noslash, max.tau = 1/zero.noslash, ...)

## S3 method for class 'profile.samediff'
plot(x, which = 1:nc, level = c(0.99, 0.95),
     fig = TRUE, ...)

## S3 method for class 'samediff'
confint(object, parm = c("tau", "delta"), level = 0.95, max = c(10, 10)
         , ...)
Arguments

fitted a samediff object
x a profile.samediff object
object a samediff object
which numeric: which parameters to profile or plot; either "1" or "2" or "1:2" to mean "tau", "delta" or both respectively.
parm the parameter(s) to compute the confidence interval for
max for profile: control parameter to specify how many units beyond the MLE, the profiling should proceed. For confint: control parameter, that can control the convergence for especially very large delta
numpts control parameter: At how many points should the profile likelihood be evaluated?
max.delta control parameter: The maximum point at which to evaluate the profile likelihood for delta
max.tau same as max.delta for "tau".
level for plot: At which levels to include horizontal lines to indicate confidence levels in plots of the normalized profile likelihoods. For confint: at which level to compute the confidence interval.
fig logical: Should the normalized profile likelihoods be plotted?
... not currently used.

Value

For profile: An object of class "profile.samediff", "data.frame"—a data.frame with two columns for each parameter profiled giving the value of the parameter and the corresponding value of the profile likelihood.

For plot: An object of class "nProfile.samediff", "data.frame"—the data.frame from the profile-object with extra columns corresponding to the which parameter containing the normalized profile likelihood.

For confint: A 2x2 matrix with columns named "lower", "upper" giving the lower and upper (1 - alpha)% confidence interval for the parameters named in the rows.

Author(s)

Rune Haubo B Christensen

See Also

summary.samediff

Examples

# data: 8 of the same samples were judged to be same
# 5 of the same samples were judged to be different
# 4 of the different samples were judged to be same
# 9 of the different samples were judged to be different

sadi <- samediff(8, 5, 4, 9)
confint(sadi)
plot(profile(sadi))
Description

Transforms or rescales estimates and optionally standard errors between the three levels at which a sensory difference is measured: pc (proportion of correct answers), pd (proportion of discriminators) and d-prime. rescale is the main function and only one of pc, pd or d-prime should be given as argument — values for the remaining two scales will be computed.

A number of auxiliary functions are also provided:

- `psyfun` implements the psychometric functions and maps from d-prime to pc
- `psyinv` implements the inverse psychometric functions and maps from pc to d-prime
- `psyderiv` implements the derivative of the psychometric functions
- `pc2pd` maps from pc to pd
- `pd2pc` maps from pd to pc

Usage

```r
rescale(pc, pd, d.prime, std.err, method = c("duotrio", "threeAFC", "twoAFC", "triangle"))
psyfun(d.prime, method = c("duotrio", "threeAFC", "twoAFC", "triangle"))
psyinv(pc, method = c("duotrio", "threeAFC", "twoAFC", "triangle"))
psyderiv(d.prime, method = c("duotrio", "threeAFC", "twoAFC", "triangle"))
pc2pd(pc, Pguess)
pd2pc(pd, Pguess)
```

Arguments

- `pc` the proportion of correct answers; a numerical vector between 0 and 1
- `pd` the proportion of discriminators; a numerical vector between 0 and 1
- `d.prime` the sensory difference on the d-prime scale; a non-negative numerical vector.
- `std.err` optional numerical vector of standard errors of the same length as the either of `pc`, `pd` or `d.prime`. Negative values are not allowed, but values may be NA
- `method` the sensory discrimination protocol for which the results should apply
- `Pguess` the guessing probability implied by the protocol; a numeric scalar between 0 and 1
Details
The rescale function is based on the fact that once the protocol and one of pc, pd and d-prime is known, the other two can be computed. The same applies to the standard errors of these parameters. Standard errors are optional, but if they are supplied, the length of the std.err argument has to match the length of pc, pd or d.prime whichever is given.
A print method is implemented for rescale objects.

Value
For rescale an object of class rescale with elements

- coefficients a data.frame with values of pc, pd and d.prime corresponding to the input
- std.err if standard errors are given through the std.err argument a data.frame of the same size and shape as coefficients with standard errors. Otherwise missing.
- method the sensory discrimination protocol for which the results apply

For psyfun, psyinv, psyderiv, pc2pd and pd2pc a numerical vector of the same length as the first argument with appropriate contents.

Author(s)
Rune Haubo B Christensen

Examples

```r
## suppose 15 out of 20 are observed in a duo-trio experiment, then
## the estimated probability of correct a answer is
(pc <- 15/20)
## The standard error of this estimate is
(se.pc <- sqrt(pc * (1 - pc) / 20))
## The corresponding estimate of proportion of discriminators (pd) and
## d-prime with associated standard errors are:
rescale(pc = pc, std.err = se.pc, method = "duotrio")

## Can also do
rescale(pd = c(.6,.7), std.err = c(.2, NA))
psyfun(2, method = "triangle")
psyinv(0.8, method = "twoAFC")
psyderiv(2, method = "duotrio")
pc2pd(0.7, 1/2)
pd2pc(0.3, 1/3)
```

ROC
Plot the Receiver Operating Characteristic Curve

Description
The function computes and plots the empirical ROC (receiver operating characteristic) curve.
Usage

ROC(object, ...)

## Default S3 method:
ROC(object, se.d, scale = 1, length = 1000,
fig = TRUE, se.type = c("CI", "SE"), CI.alpha = 0.05, ...)

## S3 method for class 'anota'
ROC(object, length = 1000, fig = TRUE,
se.type = c("CI", "SE"), CI.alpha = 0.05, ...)

Arguments

object       the class of the object defines, which of the methods is invoked. If object is a single element numeric vector it is taken as a d-prime value and the default method is invoked. If the object is of class anota, the method for anota objects is invoked.
se.d         a unit length vector with the standard error of d-prime. If supplied confidence intervals or standard errors are plotted
scale        a unit length vector giving the ratio of scale (ie. standard deviation) of the latent distribution for the no-class items relative to that of the yes-class items
length       the length of the vectors to be plotted. Longer vectors gives more smooth curves.
fig          Should a plot be produced?
se.type      The type of band for the ROC curve, "CI" for confidence interval and "SE" for standard error.
CI.alpha     the type I level of the confidence interval of AUC
...          additional arguments to plot and lines

Details

The function currently ignores the variance of the scale in the computation of the uncertainty of the ROC curve.

Value

The function makes a plot of the ROC curve, and if se.d is supplied, standard errors or confidence intervals for the curve are added to the plot.

The function also (invisibly) returns a list with the following components

ROCx          x-coordinates to the ROC curve
ROCy          y-coordinates to the ROC curve

If se.d is supplied, the object also contains

lower         y-coordinates to the lower limit
upper         y-coordinates to the upper limit

Author(s)

Rune Haubo B Christensen
Examples

```r
## ROC.default:
(mat <- matrix(c(8, 17, 1, 24), 2, byrow = TRUE))
(d.prime <- SDT(mat, "probit")[3])
ROC(d.prime)
## ROC.anota:
fm1 <- AnotA(8, 25, 1, 25)
ROC(fm1)
```

Description

Computation of tau and dprime for same different test using maximum likelihood.

Usage

`samediff(nsamesame, ndiffsame, nsamediff, ndiffdiff, VCOV = TRUE)`

Arguments

- `nsamesame`: The number of same-answers on same-samples
- `ndiffsame`: The number of different-answers on same-samples
- `nsamediff`: The number of same-answers on different-samples
- `ndiffdiff`: The number of different-answers on different-samples
- `VCOV`: Should the variance-covariance matrix of the parameters be computed. Defaults to TRUE.

Details

The function computes the maximum likelihood estimates of tau and delta.

Value

An object of class `samediff` with elements:

- `coef`: named vector of coefficients (d-prime and tau)
- `vcov`: variance-covariance matrix of the coefficients
- `se`: named vector with standard error of the coefficients (standard error of d-prime)
- `data`: a named vector with the data supplied to the function
- `test`: a string with the name of the test (same-different)
- `call`: the matched call
- `convergence`: convergence indicator. 0 indicates convergence. For error codes see optim.
- `logLik`: Value of the log-likelihood at the MLE of the parameters.
- `case`: A case indicator for internal use
samediffPwr

Author(s)
Rune Haubo B Christensen

References

Examples
# data: 8 of the same samples were judged to be same
# 5 of the same samples were judged to be different
# 4 of the different samples were judged to be same
# 9 of the different samples were judged to be different
samediff(8, 5, 4, 9)

samediffPwr

Power Analysis for Same-different Experiments

Description
Computes the power for a same-different discrimination experiment with a no-difference null hypothesis via simulation.

Usage
samediffPwr(n, tau, delta, Ns, Nd, alpha)

Arguments
n the number of samples to use in the simulation. More samples means higher precision, but takes longer to compute.
tau the value of tau
delta the underlying sensory difference under the alternative hypothesis (non-negative)
Ns the number of same-samples (a positive integer)
Nd the number of different-samples (a positive integer)
alpha the type I level of the test (must be between zero and one)

Details
The power is computed using simulations. n datasets is simulated from the Same Different model with specified parameters. The power is the fraction of times the p-value is lower than alpha.
Under some parameter combinations, there is a non-significant probability that data will fall, so that the MLE of delta is not defined and the p-value is not defined. All such undefined p-values are silently ignored.
The estimated power may change between runs and especially if the power is either very large or very small (ie. close to 0 or 1). Using more simulations will provide higher accuracy.
It is often a good idea to run the power simulation a couple of times to ensure that the variation in the result is acceptable.
samediffSim

Value
A single numeric value giving the power of the specified test.

Author(s)
Rune Haubo B Christensen

References

See Also
samediff, samediffSim

Examples
## Finding the power of a discrimination test with a sensory delta of 2
## (alternative hypothesis) versus a null hypothesis of delta = 0 with
## a sample of size 2 x 10 and a type I level of .05. n should be higher
## for a reasonable precision:
samediffPwr(n = 100, tau = 1, delta = 2, Ns = 10, Nd = 10)

samediffSim

Simulates data from a samediff test

Description
Simulates the outcome of \( n \) same-different experiments.

Usage
samediffSim(n, tau, delta, Ns, Nd)

Arguments
\( n \) the number of experiments to simulate.
\( \text{tau} \) the value of "tau".
\( \text{delta} \) the value of delta (d-prime).
\( Ns \) number of same-samples
\( Nd \) number of different-samples

Details
The function makes two calls to \texttt{rbinom}.
**SDT**

**Value**

A matrix of with \( n \) rows and four columns named \( ss, ds, sd, dd \) with the number of same-answers to same-samples, different-answers to same-samples, same-answers to different-samples and different-answers to different-samples respectively.

**Author(s)**

Rune Haubo B Christensen

**References**


**See Also**

discrimSim

**Examples**

```r
## Running simulations:
samediffSim(n = 10, tau = 1, delta = 1, Ns = 10, Nd = 10)
```

---

**SDT**

*Signal Detection Theory Computation of d-prime*

**Description**

The function computes d-prime for any 2 x J table where \( J \geq 2 \) for the "yes–no" or "A-Not A" experiment using the Signal Detection Theory (SDT) algorithm to compute \( J-1 \) d-prime’s. The algorithm is also called the "empirical probit transform". The function also provides the "logit" counterpart.

**Usage**

```r
SDT(tab, method = c("probit", "logit"))
```

**Arguments**

- `tab`: A 2 x J table with true class relation in rows (only two true classes) and the J-class response in columns
- `method`: should the empirical probit or logit transform be computed?

**Value**

A \((J-1)\times3\) matrix. The first two columns contains the z-transform of the Hit rate and the False Alarm rate respectively—ready to plot along with the empirical ROC curve. The third column contains the estimated d-primes.
Author(s)
Rune Haubo B Christensen

References

Examples

### Design table:
## 8 “yes”-responses to no-samples
## 1 “yes”-responses to yes-samples
## 17 “no”-response to no-samples
## 24 “no”-responses to yes-samples
## Note that response-class is columnwise and true-class is rowwise.
(mat <- matrix(c(8, 17, 1, 24), 2, byrow = TRUE))
SDT(mat, "logit")
SDT(mat, "probit")
## compare to AnotA():
AnotA(8, 25, 1, 25)
## Multi-response-class example (odor example from MacMillan and Creelman, 2005)
(odor <- matrix(c(112, 112, 72, 53, 22, 4, 7, 38, 0, 0, 117, 0, 62), 2, byrow = TRUE))
obj <- SDT(odor)
ROC(obj[3, 3])

summary.samediff

Summary method for samediff objects.

Description
Makes a summary of a samediff object with option to use profile likelihood for confidence intervals and p-values or the asymptotic variance-covariance matrix.

Usage

## S3 method for class 'samediff'
summary(object, profile = TRUE, ...)

Arguments

object a samediff object
profile logical: Should the profile likelihood be used for confidence intervals and p-values for the parameters? Defaults to TRUE. If FALSE the asymptotic variance-covariance matrix derived from the observed Fisher information matrix will be used. See Details for more information.
... can be level, eg 0.95 to specify the confidence level of the intervals.
threeAFC

Details

Note that the variance-covariance matrix does not always exist in contrast to the profile likelihood. `profile = FALSE` may therefore cause confidence intervals etc. to be NA.

Value

An object of class `summary.samediff` inheriting elements from the `samediff` object and with the following additional elements:

- `table` matrix with parameter estimates, standard errors, confidence intervals and p-values.
- `AIC` the AIC of the object.

Author(s)

Rune Haubo B Christensen

See Also

`confint.samediff`, `profile.samediff`

Examples

```r
# data: 8 of the same samples were judged to be same
# 5 of the same samples were judged to be different
# 4 of the different samples were judged to be same
# 9 of the different samples were judged to be different
sadi <- samediff(8, 5, 4, 9)
summary(sadi)
summary(sadi, FALSE)
```

Description

Creates a copy of the binomial family with the inverse link function changed to equal the 3-AFC psychometric function and correspondingly changed link function and derivative of the inverse link function.

Usage

```r
threeAFC()
```

Value

A binomial family object for models. Among other things it includes the psychometric function as `linkinv` and the inverse psychometric function (for direct dprime computation) as `linkfun`. 

Create 3-AFC binomial family
triangle

Create triangle binomial family

Description

Creates a copy of the binomial family with the inverse link function changed to equal the triangle psychometric function and correspondingly changed link function and derivative of the inverse link function.

Note

Several functions in this package make use of the function, but it may also be used on its own—see the example below.

Author(s)

Rune Haubo B Christensen and Per Bruun Brockhoff

References


See Also

triangle, twoAFC, duotrio, discrim, discrimPwr, discrimSim, AnotA, discrimSS, samediff, findcr

Examples

```r
xt <- matrix(c(1/zero.noslash, 5), ncol = 2) # data: 1/zero.noslash correct answers, 5 incorrect
res <- glm(xt ~ 1, family=threeAFC)
summary(res)

## Extended example plotting the profile likelihood
## data: 1/zero.noslash correct answers, 5 incorrect
xt <- matrix(c(1/zero.noslash, 2), ncol = 2)
summary(res <- glm(xt ~ 1, family = threeAFC))#, etastart = etastart))
N <- 100
dev <- double(N)
level <- c(0.95, 0.99)
delta <- seq(1e-4, 5, length = N)
for(i in 1:N)
  dev[i] <- glm(xt ~ -1 + offset(delta[i]),
    family = threeAFC)$deviance
plot(delta, exp(-dev/2), type = "l",
  xlab = expression(delta),
  ylab = "Normalized Profile Likelihood")
## Add Normal approximation:
lines(delta, exp(-(delta - coef(res))^2 /
  (2 * vcov(res))), lty = 2)
## Add confidence limits:
lim <- sapply(level, function(x)
  exp(-qchisq(x, df=1)/2)
abline(h = lim, col = "grey")
```
Usage

triangle()

Value

A binomial family object for models. Among other things it includes the psychometric function as `linkinv` and the inverse psychometric function (for direct dprime computation) as `linkfun`.

Note

Several functions in this package make use of the function, but it may also be used on its own—see the example below.

Author(s)

Rune Haubo B Christensen and Per Bruun Brockhoff

References


See Also

duotrio, twoAFC, threeAFC, discrim, discrimPwr, discrimSim, AnotA, discrimSS, samediff, findcr

Examples

```r
xt <- matrix(c(1/zero.noslash, 5), ncol = 2) ## data: 1/zero.noslash correct answers, 5 incorrect
res <- glm(xt ~ 1, family = triangle)
summary(res)

## Extended example plotting the profile likelihood
## data: 1/zero.noslash correct answers, 9 incorrect
xt <- matrix(c(1/zero.noslash, 9), ncol = 2)
sanity(res <- glm(xt ~ 1, family = triangle))
N <- 100
dev <- double(N)
delta <- seq(1e-4, 3, length = N)
for(i in 1:N)
  dev[i] <- glm(xt ~ -1 + offset(delta[i]),
                family = triangle)$deviance
plot(delta, exp(-dev/2), type = "l",
     xlab = expression(delta),
     ylab = "Normalized Profile Likelihood")
# Add Normal approximation:
lines(delta, exp(-(delta - coef(res))^2 / (2 * vcov(res))), lty = 2)
# Add confidence limits:
level <- c(0.95, 0.99)
lim <- sapply(level, function(x) exp(-qchisq(x, df=1))/2 )
abline(h = lim, col = "grey")
```
twoAC

2-AC Discrimination and Preference Protocol

Description

Computes estimates and standard errors of d-prime and tau for the two alternative (2-AC) protocol. A confidence interval and significance test for d-prime is also provided. The 2-AC protocol is equivalent to a 2-AFC protocol with a "no-difference" option, and equivalent to a paired preference test with a "no-preference" option.

Usage

twoAC(data, d.prime0 = 0, conf.level = 0.95, statistic = c("likelihood", "Wald"), alternative = c("two.sided", "less", "greater"), ...)

Arguments

data a non-negative numeric vector of length 3 with the number of observations in the three response categories in the form ("prefer A", "no-preference", "prefer B"). If the third element is larger than the first element, the estimate of d-prime is positive.
d.prime0 the value of d-prime under the null hypothesis for the significance test.
conf.level the confidence level.
statistic the statistic to use for confidence level and significance test.
alternative the type of alternative hypothesis.
... not currently used.

Details

confint, profile, logLik, vcov, and print methods are implemented for twoAC objects.

Power computations for the 2-AC protocol is implemented in twoACpwr.

Value

An object of class twoAC with elements

coefficients 2 by 2 coefficient matrix with estimates and standard errors of d-prime and tau. If the variance-covariance matrix of the parameters is not defined, the standard errors are NA.
vcov variance-covariance matrix of the parameter estimates. Only present if defined for the supplied data.
data the data supplied to the function.
call the matched call.
logLik the value of the log-likelihood at the maximum likelihood estimates.
alternative the name of the alternative hypothesis for the significance test.
statistic  
the name of the test statistic used for the significance test.

conf.level  
the confidence level for the confidence interval for d-prime.

d.prime0  
the value of d-prime under the null hypothesis in the significance test.

p.value  
p-value of the significance test.

confint  
two-sided confidence interval for d-prime. This is only available if the standard
errors are defined, which may happen in boundary cases. Use profile and
confint methods to get confidence intervals instead; see the examples.

Author(s)

Rune Haubo B Christensen

References

the 2-AC protocol. Submitted to Food Quality and Preference.

See Also

c1m2twoAC, twoACpwr

Examples

## Simple:
fit <- twoAC(c(2,2,6))
fit

## Typical discrimination-difference test:
(fit <- twoAC(data = c(2, 5, 8), d.prime0 = 0, alternative = "greater"))

## Typical discrimination-similarity test:
(fit <- twoAC(data = c(15, 15, 20), d.prime0 = .5, alternative = "less"))

## Typical preference-difference test:
(fit <- twoAC(data = c(3, 5, 12), d.prime0 = 0, alternative = "two.sided"))

## Typical preference (non-)inferiority test:
(fit <- twoAC(data = c(3, 5, 12), d.prime0 = 0, alternative = "greater"))

## For preference equivalence tests (two-sided) use CI with alpha/2:
## declare equivalence at the 5% level if 90% CI does not contain,
## e.g. -1 or 1:
(fit <- twoAC(data = c(15, 10, 10), d.prime0 = 0, conf.level = .90))

## The var-cov matrix and standard errors of the parameters are not
## defined in all situations. If standard errors are not
## defined, then confidence intervals are not provided directly:
(fit <- twoAC(c(5, 0, 15)))
## We may use profile and confint methods to get confidence intervals
## never the less:
pr <- profile(fit, range = c(-1, 3))
confint(pr)
twoACpwr

Exact Power Computation for the 2-AC Discrimination Protocol

Description

Computes the exact power for the 2-AC protocol using the (signed) likelihood root statistic. Power is computed for a significance test of d-prime. The tol argument specifies the precision with which power should be computed.

Usage

```r
twoACpwr(tau, d.prime, size, d.prime0 = 0, alpha = 0.05, tol = 1e-5, return.dist = FALSE, statistic = "likelihood", alternative = c("two.sided", "less", "greater"))
```

Arguments

- `tau`: the value of tau under the alternative hypothesis
- `d.prime`: the value of d.prime under the alternative hypothesis
- `size`: the sample size
- `d.prime0`: the value of d-prime under the null hypothesis in the significance test for which power should be computed
- `alpha`: the size of the test
- `tol`: specifies the precision with which power should be computed, e.g., 1e-4 cause power to be computed correctly to three significant digits. Lower values of tau gives higher precision, but also longer computation times.
- `return.dist`: should the p-value distribution be returned rather than the power be computed?
- `statistic`: the statistic used in the significance test for which the power should be computed. Currently only the (signed) likelihood root statistic is available—see the details for more information.
- `alternative`: the type of alternative hypothesis in the significance test for which the power should be computed

Details

The main idea in this function is to compute all possible data outcomes and then compute the p-value for the chosen significance test for each of these outcomes. This gives the exact distribution of p-values from which the exact power can be computed. This is basically what happens if tol = 0.

There is, however, a problem with this approach if size is large, since the the number of possible outcomes increases very fast with the size; the order is O(n^2). The solution to this problem is to ignore those outcomes which will occur with very small probability. Often, a large proportion of the outcomes, say 90% will occur so rarely that they account for, say 1e-4 percent of the probability mass; it is therefore safe to ignore those outcomes without compromising the accuracy of the
computed power by any relevant amount. For more information see the referenced paper and the package vignette Statistical Methodology.

The Wald statistic is not available here. The reason is that the Wald statistic is not always defined and the problem is therefore what to do with those cases where it is not defined? On the other hand the likelihood root statistic is defined in all cases, so there is no problem here, and since the likelihood root statistic is more accurate than the Wald statistic, there is not much reason to use the Wald statistic after all.

For the record; the Wald statistic is not defined, when the standard error of d-prime is not defined. This happens when the variance-covariance matrix of tau and d-prime is not defined, which occurs in a number of boundary cases, i.e., when one or more cells contain zero frequencies. Since these outcomes occur with positive probability, the algorithm used by twoACpwr will always encounter those cases and have to deal with them. This would be cumbersome to implement.

Value
A data.frame with one line and the following entries

- **power**: the computed power
- **actual.alpha**: the actual size of the test (different from the nominal alpha given as argument due to the discreteness of the observations).
- **samples**: the number of possible outcomes for this size
- **discarded**: the number of outcomes for which the p-value is not computed. This number is zero if tol = 0
- **kept**: the number of outcomes for which the p-value is computed in. This number equals samples if tol = 0
- **p**: the probability vector of the multinomial distribution implied by the values of tau and d.prime.

Author(s)
Rune Haubo B Christensen

References

See Also
clm2twoAC, twoACpwr

Examples

```r
## Exact power:
twoACpwr(tau = .5, d.prime = .7, size = 50, tol = 0)

## Power exact to a reasonable number of digits
twoACpwr(tau = .5, d.prime = .7, size = 50, tol = 1e-5)

## Power for a similarity test in a discrimination setting where the
## true parameter values are expected to be tau = 0.4 and true d.prime
## = .5, while we want to show that d.prime < 1, i.e., under the null
```
## hypothesis d.prime = 1:
twoACpwr(tau = 0.4, d.prime = -0.5, size = 100, d.prime0 = 1, tol = 1e-5, 
alternative = "less")

## Power for a difference test in a preference setting where the true
## parameter values are expected to be tau = 0.4 and d.prime = -0.5,
## while we want to show that d.prime is different from zero:
twoACpwr(tau = 0.4, d.prime = -0.5, size = 100, d.prime0 = 0, tol = 1e-5, 
alternative = "two.sided")

---

twoAFC

Create 2-AFC binomial family

**Description**

Creates a copy of the binomial family with the inverse link function changed to equal the 2-AFC
psychometric function and correspondingly changed link function and derivative of the inverse link
function.

**Usage**

twoAFC()

**Value**

A binomial family object for models. Among other things it includes the psychometric function as
`linkinv` and the inverse psychometric function (for direct dprime computation) as `linkfun`.

**Note**

Several functions in this package make use of the function, but it may also be used on its own—see
the example below.

**Author(s)**

Rune Haubo B Christensen and Per Bruun Brockhoff

**References**


**See Also**

triangle, threeAFC, duotrio, discrim, discrimPwr, discrimSim, AnotA, discrimSS, samediff,
findcr
Examples

xt <- matrix(c(1/zero.noslash, 5), ncol = 2) ## data: 1/zero.noslash correct answers, 5 incorrect
res <- glm(xt ~ 1, family = twoAFC)
summary(res)

## Extended example plotting the profile likelihood
## data: 1/zero.noslash correct and 8 incorrect:
xt <- matrix(c(1/zero.noslash, 8), ncol = 2)
summary(res <- glm(xt ~ 1, family = twoAFC))
N <- 100
dev <- double(N)
level <- c(0.95, 0.99)
delta <- seq(1e-4, 3, length = N)
for(i in 1:N)
  dev[i] <- glm(xt ~ -1 + offset(delta[i]),
                family = twoAFC)$deviance
plot(delta, exp(-dev/2), type = "l",
     xlab = expression(delta),
     ylab = "Normalized Profile Likelihood")

## Add Normal approximation:
lines(delta, exp(-(delta - coef(res))^2 / 
      (2 * vcov(res))), lty = 2)

## Add confidence limits:
lim <- sapply(level, function(x)
  exp(-qchisq(x, df=1)/2) )
abline(h = lim, col = "grey")
Index

+Topic **hplot**
  - plot.discrim, 24
  - plot.samediff, 25
  - ROC, 30

+Topic **htest**
  - AUC, 4

+Topic **models**
  - Anota, 2
  - betabin, 5
  - clls, 8
  - clm2twoAC, 10
  - confint.twoAC, 12
  - discrim, 14
  - discrimPwr, 16
  - discrimR, 17
  - discrimSim, 19
  - discrimSS, 20
  - duotrio, 22
  - findcr, 23
  - profile.discrim, 26
  - profile.samediff, 27
  - rescale, 29
  - samediff, 32
  - samediffPwr, 33
  - samediffSim, 34
  - SDT, 35
  - summary.samediff, 36
  - threeAFC, 37
  - triangle, 38
  - twoAC, 40
  - twoACpwr, 42
  - twoAFC, 44

AnotA, 2, 4, 15, 17, 19–22, 24, 38, 39, 44
anova, 9
AUC, 4

beta, 7
betabin, 5

c1ls, 8
clm, 10, 11
clm2twoAC, 10, 41, 43
c1mm, 10, 11

c1nt, 13, 15, 40
confint.anota (Anota), 2
confint.discrim (profile.discrim), 26
confint.profile.twoAC (confint.twoAC), 12
confint.samediff, 37
confint.samediff (profile.samediff), 27
confint.twoAC, 12
d.primePwr (discrimPwr), 16
d.primeSS (discrimSS), 20
discrim, 3, 14, 17, 20, 22, 24, 26, 27, 38, 39, 44
discrimPwr, 3, 15, 16, 19–22, 24, 38, 39, 44
discrimR, 17
discrimSim, 3, 15, 17, 19, 22, 24, 35, 38, 39, 44
discrimSS, 3, 15, 17, 19, 22, 24, 38, 39, 44
duotrio, 7, 15, 19, 20, 22, 24, 38, 39, 44
findcr, 3, 15, 17, 19–22, 23, 38, 39, 44
formula, 8
glm, 10, 18
multinom, 10

optim, 7, 9, 10, 18, 32

pc2pd (rescale), 29
pd2pc (rescale), 29
plot.anota (Anota), 2
plot.discrim, 24
plot.profile, 15
plot.profile.discrim (profile.discrim), 26
plot.profile.samediff (profile.samediff), 27
plot.profile.twoAC (confint.twoAC), 12
plot.samediff, 25
polr, 9, 10
print.AUC (AUC), 4
print.discrim, 3
print.discrim (discrim), 14

46
INDEX

print.twoAC (twoAC), 40
profile, 13, 15, 40
profile.discrim, 26
profile.samediff, 27, 37
profile.twoAC (confint.twoAC), 12
prop.test, 15
psyderiv (rescale), 29
psyfun (rescale), 29
psyinv (rescale), 29
rbinom, 34
rescale, 29
ROC, 30
samediff, 15, 19–22, 24, 32, 34, 38, 39, 44
samediffPwr, 33
samediffSim, 34, 34
SDT, 35
spline, 26
summary, 9
summary.betabin (betabin), 5
summary.samediff, 28, 36
threeAFC, 7, 15, 19, 20, 22, 24, 37, 39, 44
triangle, 7, 15, 19, 20, 22, 24, 38, 38, 44
twoAC, 11, 12, 40
twoACpwr, 11, 40, 41, 42, 43
twoAFC, 7, 15, 19, 20, 22, 24, 38, 39, 44
vcov, 9
Appendix J

Analysis of ordinal data with cumulative link models — estimation with the R package ordinal

Analysis of ordinal data with cumulative link models — estimation with the R-package \texttt{ordinal}

Rune Haubo B Christensen

September 25, 2012
# Contents

1 Introduction 3

2 Cumulative link models 4
   2.1 Fitting cumulative link models with clm from package ordinal 6
   2.2 Odds ratios and proportional odds 8
   2.3 Link functions 9
   2.4 Maximum likelihood estimation of cumulative link models 10
   2.5 Deviance and model comparison 12
      2.5.1 Model comparison with likelihood ratio tests 12
      2.5.2 Deviance and ANODEV tables 12
      2.5.3 Goodness of fit tests with the deviance 14
   2.6 Latent variable motivation for cumulative link models 15
      2.6.1 More on parameter interpretation 17
   2.7 Structured thresholds 19
      2.7.1 Symmetric thresholds 20
      2.7.2 Equidistant thresholds 20

3 Assessing the likelihood and model convergence 22

4 Confidence intervals and profile likelihood 25

A An approximate ML estimation method for CLMs 28
1 Introduction

Ordered categorical data, or simply ordinal data, are commonplace in scientific disciplines where humans are used as measurement instruments. Examples include school gradings, ratings of preference in consumer studies, degree of tumor involvement in MR images and animal fitness in field ecology. Cumulative link models are a powerful model class for such data since observations are treated rightfully as categorical, the ordered nature is exploited and the flexible regression framework allows in-depth analyses.

The name cumulative link models is adopted from Agresti (2002), but the models are also known as ordinal regression models although that term is sometimes also used for other regression models for ordinal responses such as continuation ratio models (see e.g., Agresti, 2002). Other aliases are ordered logit models and ordered probit models (Greene and Hensher, 2010) for the logit and probit link functions. Further, the cumulative link model with a logit link is widely known as the proportional odds model due to McCullagh (1980), also with a complementary log-log link, the model is known as proportional hazards model for grouped survival times.

Ordinal response variables can be analyzed with omnibus Pearson $\chi^2$ tests, base-line logit models or log-linear models. This corresponds to assuming that the response variable is nominal and information about the ordering of the categories will be ignored. Alternatively numbers can be attached to the response categories, e.g., 1, 2, ..., $J$ and the resulting scores can be analyzed by conventional linear regression and ANOVA models. This approach is in a sense over-confident since the data are assumed to contain more information than they actually do. Observations on an ordinal scale are classified in ordered categories, but the distance between the categories is generally unknown. By using linear models the choice of scoring impose assumptions about the distance between the response categories. Further, standard errors and tests from linear models rest on the assumption that the response, conditional on the explanatory variables, is normally distributed (equivalently the residuals are assumed to be normally distributed). This cannot be the case since the scores are discrete and responses beyond the end categories are not possible. If there are many responses in the end categories, there will most likely be variance heterogeneity to which $F$ and $t$ tests can be rather sensitive. If there are many response categories and the response does not pile up in the end categories, we may expect tests from linear models to be accurate enough, but any bias and optimism is hard to quantify.

Cumulative link models provide the regression framework familiar from linear models while treating the response rightfully as categorical. While cumulative link models are not the only type of ordinal regression model, they are by far the most popular class of ordinal regression models.

Common to the application of methods for nominal responses and linear models to ordinal responses is that interpretation of effects on the ordinal response scale is awkward. For example, linear models will eventually give predictions outside the possible range and statements such as “the response increase 1.2 units with each degree increase in temperature” will only be approximately valid in a restricted range of the response variable.

In this document cumulative link models are described for modeling ordinal response variables. We also describe these models are fitted and data are analyzed with the functionality provided in the ordinal package for R (R Development Core Team, 2011).

Example 1 (The wine data): As an example of the data set with an ordinal response variable
consider the wine data from Randall (1989) available in the object \texttt{wine} in package \texttt{ordinal}, cf. Table 1. The data represent a factorial experiment on factors determining the bitterness of wine with 1 = “least bitter” and 5 = “most bitter”. Two treatment factors (temperature and contact) each have two levels. Temperature and contact between juice and skins can be controlled when crushing grapes during wine production. Nine judges each assessed wine from two bottles from each of the four treatment conditions, hence there are 72 observations in all. In Table 1 we have aggregated data over bottles and judges for simplicity, but these variables will be considered later. Initially we only assume that, say, category 4 is larger than 3, but not that the distance between 2 and 3 is half the distance between 2 and 4, for example. The main objective is to examine the effect of contact and temperature on the perceived bitterness of wine.

2 Cumulative link models

A cumulative link model is a model for an ordinal response variable, \(Y_i\) that can fall in \(j = 1, \ldots, J\) categories.\(^1\) Then \(Y_i\) follows a multinomial distribution with parameter \(\pi\) where \(\pi_{ij}\) denote the probability that the \(i\)th observation falls in response category \(j\). We define the cumulative probabilities as:\(^2\)

\[
\gamma_{ij} = P(Y_i \leq j) = \pi_{i1} + \ldots + \pi_{ij}. \tag{1}
\]

Initially we will consider the logit link. The logit function is defined as \(\text{logit}(\pi) = \log[\pi/(1 − \pi)]\) and cumulative logits are defined as:

\[
\text{logit}(\gamma_{ij}) = \log(P(Y_i \leq j)) = \log\left(\frac{P(Y_i \leq j)}{1 − P(Y_i \leq j)}\right) \quad j = 1, \ldots, J − 1 \tag{2}
\]

so that the cumulative logits are defined for all but the last category.\(^3\)

A cumulative link model with a logit link, or simply \textit{cumulative logit model} is a regression model for cumulative logits:

\[
\text{logit}(\gamma_{ij}) = \theta_j - \mathbf{x}_i^T \beta \tag{3}
\]

where \(\mathbf{x}_i\) is a vector of explanatory variables for the \(i\)th observation and \(\beta\) is the corresponding set of regression parameters. The \(\{\theta_j\}\) parameters provide each cumulative logit (for each \(j\)) with its own intercept. A key point is that the regression part \(\mathbf{x}_i^T \beta\) is independent of \(j\), so \(\beta\) has the same effect for each of the \(J − 1\) cumulative logits. Note that \(\mathbf{x}_i^T \beta\) does

\(^{1}\)where \(J \geq 2\). If \(J = 2\) binomial models also apply, and in fact the cumulative link model is in this situation identical to a generalized linear model for a binomial response.

\(^{2}\)we have suppressed the conditioning on the covariate vector, \(\mathbf{x}_i\), so we have that \(\gamma_{ij} = \gamma_j(\mathbf{x}_i)\) and \(P(Y_i \leq j) = P(Y \leq j | \mathbf{x}_i)\).

\(^{3}\)since for \(j = J\) the denominator would be \(1 − P(Y_i \leq J) = 1 − 1 = 0\) and thus the fraction is not defined.
\[ \gamma_i = P(Y_i \leq j) \]
\[ j = 1, 2, 3 \]

Figure 1: Illustration of a cumulative link model with four response categories.

Table 2: Estimates from ordinary logistic regression models (OLR) and a cumulative logit model (CLM) fitted to the wine data (cf. Table 1).

<table>
<thead>
<tr>
<th>j</th>
<th>OLR</th>
<th>CLM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intercept</td>
<td>$\theta_j$</td>
</tr>
<tr>
<td>1</td>
<td>-2.08</td>
<td>-1.48</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>-1.10</td>
</tr>
<tr>
<td>3</td>
<td>1.82</td>
<td>-1.37</td>
</tr>
<tr>
<td>4</td>
<td>2.83</td>
<td>-1.01</td>
</tr>
</tbody>
</table>

not contain an intercept, since the $\{\theta_j\}$ act as intercepts. The cumulative logit model is illustrated in Fig. 1 for data with four response categories. For small values of $x_i^T \beta$ the response is likely to fall in the first category and for large values of $x_i^T \beta$ the response is likely to fall in the last category. The horizontal displacements of the curves are given by the values of $\{\theta_j\}$.

Some sources write the cumulative logit model, (3) with a plus on the right-hand-side, but there are two good reasons for the minus. First, it means that the larger the value of $x_i^T \beta$, the higher the probability of the response falling in a category at the upper end of the response scale. Thus $\beta$ has the same direction of effect as the regression parameter in an ordinary linear regression or ANOVA model. The second reason is related to the latent variable interpretation of cumulative link models that we will consider in section 2.6.

**Example 2:** We will consider a cumulative logit model for the wine data including an effect of contact. The response variable is the bitterness ratings, thus each $y_i$ takes a value from 1 to 5 representing the degree of bitterness for the $i$th sample. We may write the cumulative logit model as:

\[ \logit(\gamma_{ij}) = \theta_j - \beta_2(\text{contact}_i), \quad j = 1, \ldots, 4, \quad i = 1, \ldots, 72. \]

The parameter estimates are given in the last two columns of Table 2. If we consider the cumulative logit model (3) for a fixed $j$, e.g., for $j = 1$, then the model is just an ordinary logistic regression model where the binomial response is divided into those observations falling in category $j$ or less ($Y_i \leq j$), and those falling in a higher category than $j$ ($Y_i > j$). An initial analysis might indeed start by fitting such an ordinary logistic regression models for a dichotomized response. If we proceed in this vein, we could fit the $J - 1 = 4$ ordinary logistic
regression model by fixing \( j \) at 1, 2, 3, and 4 in turn. The estimates of those four ordinary logistic regression models are given in Table 2 under the OLR heading. The cumulative logit model can be seen as the model that combines these four ordinary logistic regression models into a single model and therefore makes better use of the information in the data. A part from a sign difference the estimate of the effect of contact from the cumulative logit model is about the average of the contact effect estimates from the ordinary logistic regression models. Also observe that the standard error of the contact effect is smaller in the cumulative logit model than in any of the ordinary logistic regression models reflecting that the effect of contact is more accurately determined in the cumulative logit model. The intercepts from the ordinary logistic regression models are also seen to correspond to the threshold parameters in the cumulative logit model.

The four ordinary logistic regression models can be combined in a single ordinary logistic regression model providing an even better approximation to the cumulative link model. This is considered further in Appendix A.

2.1 Fitting cumulative link models with \texttt{clm} from package \texttt{ordinal}

Cumulative link models can be fitted with \texttt{clm} from package \texttt{ordinal}. The function takes the following arguments:

\begin{verbatim}
function (formula, scale, nominal, data, weights, start, subset, doFit = TRUE,
na.action, contrasts, model = TRUE, control = list(), link = c("logit",
"probit", "cloglog", "loglog", "cauchit"), threshold = c("flexible",
"symmetric", "equidistant"), ...)  
\end{verbatim}

Most arguments are standard and well-known from \texttt{lm} and \texttt{glm}, so they will not be introduced. The \texttt{formula} argument is of the form \texttt{response ~ covariates} and specifies the linear predictor. The \texttt{response} should be an \texttt{ordered factor} (see \texttt{help(factor)}) with levels corresponding to the response categories. A number of link functions are available and the logit link is the default. The \texttt{doFit} and \texttt{threshold} arguments will be introduced in later sections. For further information about the arguments see the help page for \texttt{clm}.

\textbf{Example 3:} In this example we fit a cumulative logit model to the wine data presented in example 1 with \texttt{clm} from package \texttt{ordinal}. A cumulative logit model that includes additive effects of temperature and contact is fitted and summarized with

\begin{verbatim}
> fm1 <- clm(rating ~ contact + temp, data = wine)
> summary(fm1)
\end{verbatim}

\texttt{formula: rating ~ contact + temp}

\texttt{data: wine}

\texttt{link flexible 72 -86.49 184.98 6(0) 4.02e-12 2.7e+01}

\texttt{Coefficients:}

\begin{verbatim}
        Estimate Std. Error z value Pr(>|z|)
contactyes  1.5278    0.4766  3.205  0.00135 **
tempwarm    2.5031    0.5287  4.735  2.19e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
\end{verbatim}

\footnote{Typing \texttt{?clm} or \texttt{help(clm)} in the command prompt should display the help page for \texttt{clm}.}
Threshold coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.3444</td>
<td>0.5171</td>
<td>-2.600</td>
</tr>
<tr>
<td>2</td>
<td>1.2508</td>
<td>0.4379</td>
<td>2.857</td>
</tr>
<tr>
<td>3</td>
<td>3.4669</td>
<td>0.5978</td>
<td>5.800</td>
</tr>
<tr>
<td>4</td>
<td>5.0064</td>
<td>0.7309</td>
<td>6.850</td>
</tr>
</tbody>
</table>

The summary provides basic information about the model fit. There are two coefficient tables: one for the regression variables and one for the thresholds or cut-points. Often the thresholds are not of primary interest, but they are an integral part of the model. It is not relevant to test whether the thresholds are equal to zero, so no p-values are provided for this test. The condition number of the Hessian is a measure of how identifiable the model is; large values, say larger than 1e4 indicate that the model may be ill defined. From this model it appears that contact and high temperature both lead to higher probabilities of observations in the high categories as we would also expect from examining Table 1.

The Wald tests provided by `summary` indicate that both contact and temperature effects are strong. More accurate likelihood ratio tests can be obtained using the `drop1` and `add1` methods (equivalently `dropterm` or `addterm`). The Wald tests are marginal tests so the test of e.g., `temp` is measuring the effect of temperature while controlling for the effect of contact. The equivalent likelihood ratio tests are provided by the `drop`-methods:

```R
> drop1(fm1, test = "Chi")
```

### Single term deletions

Model:
```
rating ~ contact + temp
df AIC LRT Pr(>Chi)
<none> 184.98
contact 1 194.03 11.043 0.0008902 ***
temp 1 209.91 26.928 2.112e-07 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case the likelihood ratio tests are slightly more significant than the Wald tests. We could also have tested the effect of the variables while ignoring the effect of the other variable. For this test we use the `add`-methods:

```R
> fm0 <- clm(rating ~ 1, data = wine)
> add1(fm0, scope = ~ contact + temp, test = "Chi")
```

### Single term additions

Model:
```
rating ~ 1
df AIC LRT Pr(>Chi)
<none> 215.44
contact 1 209.91 7.5263 0.00608 **
temp 1 194.03 23.4113 1.308e-06 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

where we used the `scope` argument to indicate which terms to include in the model formula. These tests are a little less significant than the tests controlling for the effect of the other variable.

Conventional symmetric so-called Wald confidence intervals for the parameters are available as

```R
> confint(fm1, type = "Wald")
```
2.2 Odds ratios and proportional confidence odds

The odds ratio of the event \( Y \leq j \) at \( x_1 \) relative to the same event at \( x_2 \) is

\[
\text{OR} = \frac{\gamma_j(x_1)/[1 - \gamma_j(x_1)]}{\gamma_j(x_2)/[1 - \gamma_j(x_2)]} = \exp(\theta_j - x_1^T \beta) / \exp(\theta_j - x_2^T \beta) = \exp[(x_2^T - x_1^T) \beta]
\]

which is independent of \( j \). Thus the cumulative odds ratio is proportional to the distance between \( x_1 \) and \( x_2 \) which made McCullagh (1980) call the cumulative logit model a proportional odds model. If \( x \) represent a treatment variable with two levels (e.g., placebo and treatment), then \( x_2 - x_1 = 1 \) and the odds ratio is \( \exp(-\beta_{\text{treatment}}) \). Similarly the odds ratio of the event \( Y \geq j \) is \( \exp(\beta_{\text{treatment}}) \).

Confidence intervals for the odds ratios are obtained by transforming the limits of confidence intervals for \( \beta \), which will lead to asymmetric confidence intervals for the odds ratios. Symmetric confidence intervals constructed from the standard error of the odds ratios will not be appropriate and should be avoided.

**Example 4:** The (cumulative) odds ratio of rating \( \geq j \) (for all \( j = 1, \ldots, J - 1 \)) for contact and temperature are

\[
\begin{align*}
&> \text{round}(\exp(\text{fm1$beta}), 1) \\
& \text{contactyes} \ & \text{tempwarm} \\
& 4.6 \ & 12.2
\end{align*}
\]

attesting to the strong effects of contact and temperature. Asymmetric confidence intervals for the odds ratios based on the Wald statistic are:

\[
\begin{align*}
&> \text{round}(\exp(\text{confint(fm1, type = "Wald")}, 1) \\
& 2.5 \% \ & 97.5 \% \\
& 1|2 \ & 0.1 \ & 0.7 \\
& 2|3 \ & 1.5 \ & 8.2 \\
& 3|4 \ & 9.9 \ & 103.4 \\
& 4|5 \ & 35.7 \ & 625.8 \\
& \text{contactyes} \ & 1.8 \ & 11.7 \\
& \text{tempwarm} \ & 4.3 \ & 34.4
\end{align*}
\]
<table>
<thead>
<tr>
<th>Name</th>
<th>logit</th>
<th>probit</th>
<th>log-log</th>
<th>clog-log</th>
<th>cauchit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution</td>
<td>logistic</td>
<td>Normal</td>
<td>Gumbel (max)(^b)</td>
<td>Gumbel (min)(^b)</td>
<td>Cauchy(^c)</td>
</tr>
<tr>
<td>Shape</td>
<td>symmetric</td>
<td>symmetric</td>
<td>right skew</td>
<td>left skew</td>
<td>kurtotic</td>
</tr>
<tr>
<td>Link function ((F^{-1}))</td>
<td>log[(\gamma/(1 - \gamma))]</td>
<td>(\Phi^{-1}(\gamma))</td>
<td>(-\log[-\log(\gamma)])</td>
<td>(\log[-\log(1 - \gamma)])</td>
<td>tan((\pi(\gamma - 0.5)))</td>
</tr>
<tr>
<td>Inverse link ((F))</td>
<td>(1/[1 + \exp(\eta)])</td>
<td>(\Phi(\eta))</td>
<td>(\exp(-\exp(-\eta)))</td>
<td>(1 - \exp[-\exp(\eta)])</td>
<td>(\arctan(\eta)/\pi + 0.5)</td>
</tr>
<tr>
<td>Density ((f = F'))</td>
<td>(\exp(-\eta)/[1 + \exp(-\eta)])</td>
<td>(\phi(\eta))</td>
<td>(\exp(-\exp(-\eta) - \eta))</td>
<td>(\exp(-\exp(\eta) + \eta))</td>
<td>(1/[\pi(1 + \eta^2)])</td>
</tr>
</tbody>
</table>

\(^a\): the complementary log-log link  
\(^b\): the Gumbel distribution is also known as the extreme value (type I) distribution for extreme minima or maxima. It is also sometimes referred to as the Weibull (or log-Weibull) distribution (http://en.wikipedia.org/wiki/Gumbel_distribution).  
\(^c\): the Cauchy distribution is a \(t\)-distribution with one df

### 2.3 Link functions

Cumulative link models are not formally a member of the class of (univariate) generalized linear models\(^5\) (McCullagh and Nelder, 1989), but they share many similarities with generalized linear models. Notably a link function and a linear predictor (\(\eta_j = \theta_j - x_i^T \beta\)) needs to be specified as in generalized linear models while the response distribution is just the multinomial. Fahrmeir and Tutz (2001) argues that cumulative link models are members of a class of multivariate generalized linear models. In addition to the logit link other choices are the probit, cauchit, log-log and clog-log links. These are summarized in Table 3. The cumulative link model may be written as

\[
\gamma_{ij} = F(\eta_{ij}), \quad \eta_{ij} = \theta_j - x_i^T \beta
\]  

where \(F^{-1}\) is the link function—the motivation for this particular notation will be given in section 2.6.

The probit link is often used when the model is interpreted with reference to a latent variable, cf. section 2.6. When the response variable represent grouped duration or survival times the complementary log-log link is often used. This leads to the proportional hazard model for grouped responses:

\[
-\log\{1 - \gamma_j(x_i)\} = \exp(\theta_j - x_i^T \beta)
\]

or equivalently

\[
\log[-\log\{1 - \gamma_j(x_i)\}] = \theta_j - x_i^T \beta.
\]  

Here \(1 - \gamma_j(x_i)\) is the probability or survival beyond category \(j\) given \(x_i\). The proportional hazards model has the property that

\[
\log\{\gamma_j(x_1)\} = \exp[(x_2^T - x_1^T)\beta] \log\{\gamma_j(x_2)\}.
\]

If the log-log link is used on the response categories in the reverse order, this is equivalent to using the c-log-log link on the response in the original order. This reverses the sign of \(\beta\) as well as the sign and order of \(\{\theta_j\}\) while the likelihood and standard errors remain unchanged.

---

\(^5\)the distribution of the response, the multinomial, is not a member of the (univariate) exponential family of distributions.
Table 4: Income distribution (percentages) in the Northeast US adopted from McCullagh (1980).

<table>
<thead>
<tr>
<th>Year</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-3</td>
</tr>
<tr>
<td>1960</td>
<td>6.50</td>
</tr>
<tr>
<td>1970</td>
<td>4.30</td>
</tr>
</tbody>
</table>

In addition to the standard links in Table 3, flexible link functions are available for `clm` in package `ordinal`.

**Example 5:** McCullagh (1980) present data on income distribution in the Northeast US reproduced in Table 4 and available in package `ordinal` as the object `income`. The unit of the income groups are thousands of (constant) 1973 US dollars. The numbers in the body of the table are percentages of the population summing to 100 in each row\(^6\), so these are not the original observations. The uncertainty of parameter estimates depends on the sample size, which is unknown here, so we will not consider hypothesis tests. Rather the most important systematic component is an upward shift in the income distribution from 1960 to 1970 which can be estimated from a cumulative link model. This is possible since the parameter estimates themselves only depend on the relative proportions and not the absolute numbers.

McCullagh considers which of the logit or cloglog links best fit the data in a model with an additive effect of `year`. He concludes that a the complementary log-log link corresponding to a right-skew distribution is a good choice. We can compare the relative merit of the links by comparing the value of the log-likelihood of models with different link functions:

```r
> links <- c("logit", "probit", "cloglog", "loglog", "cauchit")
> sapply(links, function(link) {
  clm(income ~ year, data=income, weights=pct, link=link)$logLik })
```

<table>
<thead>
<tr>
<th>Link</th>
<th>Log-Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>logit</td>
<td>-353.3589</td>
</tr>
<tr>
<td>probit</td>
<td>-353.8036</td>
</tr>
<tr>
<td>cloglog</td>
<td>-352.8980</td>
</tr>
<tr>
<td>loglog</td>
<td>-355.6028</td>
</tr>
<tr>
<td>cauchit</td>
<td>-352.8434</td>
</tr>
</tbody>
</table>

The cauchit link attains the highest log-likelihood closely followed by the complementary log-log link. This indicates that a symmetric heavy tailed distribution such as the Cauchy provides an even slightly better description of these data than a right skew distribution.

Adopting the complementary log-log link we can summarize the connection between the income in the two years by the following: If \( p_{1960}(x) \) is proportion of the population with an income larger than \( x \) in 1960 and \( p_{1970}(x) \) is the equivalent in 1970, then approximately

\[
\log p_{1960}(x) = \exp(\hat{\beta}) \log p_{1970}(x)
\]

\[= \exp(0.568) \log p_{1970}(x)\]

\[\square\]

### 2.4 Maximum likelihood estimation of cumulative link models

Cumulative link models are usually estimated by maximum likelihood (ML) and this is also the criterion used in package `ordinal`. The log-likelihood function (ignoring additive constants) can be written as

\[
\ell(\theta, \beta; y) = \sum_{i=1}^{n} w_i \log \pi_i
\]

\[^6\text{save rounding error}\]
where \(i\) index all scalar observations (not multinomial vector observations), \(w_i\) are potential case weights and \(\pi_i\) is the probability of the \(i\)th observation falling in the response category that it did, i.e., \(\pi_i\) are the non-zero elements of \(\pi_{ij} I(Y_i = j)\). Here \(I(\cdot)\) is the indicator function being 1 if its argument is true and zero otherwise. The ML estimates of the parameters; \(\hat{\theta}\) and \(\hat{\beta}\) are those values of \(\theta\) and \(\beta\) that maximize the log-likelihood function in (6).

Not all data sets can be summarized in a table like Table 1. If a continuous variable takes a unique value for each observation, each row of the resulting table would contain a single 1 and zeroes for the rest. In this case all \(\{w_i\}\) are one unless the observations are weighted for some other reason. If the data can be summarized as in Table 1, a multinomial observation vector such as \([3, 1, 2]\) can be fitted using \(y = [1, 1, 1, 2, 3, 3]\) with \(w = [1, 1, 1, 1, 1, 1]\) or by using \(y = [1, 2, 3]\) with \(w = [3, 1, 2]\). The latter construction is considerably more computationally efficient (and therefore faster) since the log-likelihood function contains three rather than six terms and the design matrix, \(X\) will have three rather than six rows.

The details of the actual algorithm by which the likelihood function is optimized is deferred to a later section.

According to standard likelihood theory, the variance-covariance matrix of the parameters can be obtained as the inverse of the observed Fisher information matrix. This matrix is given by the negative Hessian of the log-likelihood function\(^7\) evaluated at the maximum likelihood estimates. Standard errors can be obtained as the square root of the diagonal of the variance-covariance matrix.

Let \(\alpha = [\theta, \beta]\) denote the full set of parameters. The Hessian matrix is then given as the second order derivative of the log-likelihood function evaluated at the ML estimates:

\[
H = \frac{\partial^2 l(\alpha; y)}{\partial \alpha \partial \alpha^T} \bigg|_{\alpha = \hat{\alpha}}.
\]

The observed Fisher information matrix is then \(I(\hat{\alpha}) = -H\) and the standard errors are given by

\[
se(\hat{\alpha}) = \sqrt{\text{diag}[I(\hat{\alpha})^{-1}]} = \sqrt{\text{diag}[-H(\hat{\alpha})^{-1}]}.\]

Another general way to obtain the variance-covariance matrix of the parameters is to use the expected Fisher information matrix. The choice of whether to use the observed or the expected Fisher information matrix is often dictated by the fitting algorithm: re-weighted least squares methods often produce the expected Fisher information matrix as a by-product of the algorithm, and Newton-Raphson algorithms (such as the one used for \texttt{clm} in \texttt{ordinal}) similarly produce the observed Fisher information matrix. Efron and Hinkley (1978) considered the choice of observed versus expected Fisher information and argued that the observed information contains relevant information thus it is preferred over the expected information.

Pratt (1981) and Burridge (1981) showed (seemingly independent of each other) that the log-likelihood function of cumulative link models with the link functions considered in Table 3, except for the cauchit link, is concave. This means that there is a unique global optimum so there is no risk of convergence to a local optimum. It also means that the step of a Newton-Raphson algorithm is guaranteed to be in the direction of a higher likelihood although the step may be too large to cause an increase in the likelihood. Successively halving the step whenever this happens effectively ensures convergence.

\(^7\)equivalently the Hessian of the negative log-likelihood function.
Notably the log likelihood of cumulative cauchit models is not guarantied to be concave, so convergence problems may occur with the Newton-Raphson algorithm. Using the estimates from a cumulative probit models as starting values seems to be a widely successful approach. Observe also that the concavity property does not extend to cumulative link models with scale effects, but that structured thresholds (cf. section 2.7) are included.

2.5 Deviance and model comparison

2.5.1 Model comparison with likelihood ratio tests

A general way to compare models is by means of the likelihood ratio statistic. Consider two models, $m_0$ and $m_1$, where $m_0$ is a submodel of model $m_1$, that is, $m_0$ is simpler than $m_1$ and $m_0$ is nested in $m_1$. The likelihood ratio statistic for the comparison of $m_0$ and $m_1$ is

$$LR = -2(\ell_0 - \ell_1)$$

where $\ell_0$ is the log-likelihood of $m_0$ and $\ell_1$ is the log-likelihood of $m_1$. The likelihood ratio statistic measures the evidence in the data for the extra complexity in $m_1$ relative to $m_0$. The likelihood ratio statistic asymptotically follows a $\chi^2$ distribution with degrees of freedom equal to the difference in the number of parameter of $m_0$ and $m_1$. The likelihood ratio test is generally more accurate than Wald tests. Cumulative link models can be compared by means of likelihood ratio tests with the anova method.

Example 6: Consider the additive model for the wine data in example 3 with a main effect of temperature and contact. We can use the likelihood ratio test to assess whether the interaction between these factors are supported by the data:

```r
> fm2 <- clm(rating ~ contact * temp, data = wine)
> anova(fm1, fm2)
```

<table>
<thead>
<tr>
<th>formula:</th>
<th>link:</th>
<th>threshold:</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm1 rating ~ contact + temp logit flexible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fm2 rating ~ contact * temp logit flexible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>no.par</th>
<th>AIC</th>
<th>logLik LR.stat df Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm1</td>
<td>6</td>
<td>184.98 -86.492</td>
</tr>
<tr>
<td>fm2</td>
<td>7</td>
<td>186.83 -86.416 0.1514 1 0.6972</td>
</tr>
</tbody>
</table>

The likelihood ratio statistic is small in this case and compared to a $\chi^2$ distribution with 1 df, the $p$-value turns out insignificant. We conclude that the interaction is not supported by the data.

2.5.2 Deviance and ANODEV tables

In linear models ANOVA tables and $F$-tests are based on the decomposition of sums of squares. The concept of sums of squares does not make much sense for categorical observations, but a more general measure called the deviance is defined for generalized linear models and contingency tables\(^8\). The deviance can be used in much the same way to compare nested

---

\(^{8}\)i.e., for likelihood based models for contingency tables
models and to make a so-called analysis of deviance (ANODEV) table. The deviance is closely related to sums of squares for linear models (McCullagh and Nelder, 1989).

The deviance is defined as minus twice the difference between the log-likelihoods of a full (or saturated) model and a reduced model:

$$D = -2(\ell_{\text{reduced}} - \ell_{\text{full}})$$  \hspace{1cm} (10)

The full model has a parameter for each observation and describes the data perfectly while the reduced model provides a more concise description of the data with fewer parameters.

A special reduced model is the null model which describes no other structure in the data than what is implied by the design. The corresponding deviance is known as the null deviance and analogous to the total sums of squares for linear models. The null deviance is therefore also denoted the total deviance. The residual deviance is a concept similar to a residual sums of squares and simply defined as

$$D_{\text{resid}} = D_{\text{total}} - D_{\text{reduced}}$$  \hspace{1cm} (11)

A difference in deviance between two nested models is identical to the likelihood ratio statistic for the comparison of these models. Thus the deviance difference, just like the likelihood ratio statistic, asymptotically follows a $\chi^2$-distribution with degrees of freedom equal to the difference in the number of parameters in the two models. In fact the deviance in (10) is just the likelihood ratio statistic for the comparison of the full and reduced models.

The likelihood of reduced models are available from fits of cumulative link models, but since it is not always easy to express the full model as a cumulative link model, the log-likelihood of the full model has to be obtained in another way. For a two-way table like Table 1 indexed by $h$ (rows) and $j$ (columns), the log-likelihood of the full model (comparable to the likelihood in (6)) is given by

$$\ell_{\text{full}} = \sum_h \sum_j w_{hj} \log \hat{\pi}_{hj}$$  \hspace{1cm} (12)

where $\hat{\pi}_{hj} = w_{hj} / w_h$, $w_{hj}$ is the count in the $(h,j)$th cell and $w_h$ is the sum in row $h$.

**Example 7:** We can get the likelihood of the full model for the wine data in Table 1 with

```r
> tab <- with(wine, table(temp:contact, rating))
> ## Get full log-likelihood:
> pi.hat <- tab / rowSums(tab)
> (ll.full <- sum(tab * ifelse(pi.hat > 0, log(pi.hat), 0))) ## -84.01558
[1] -84.01558
```

The total deviance (10) for the wine data is given by

```r
> ## fit null-model:
> fm0 <- clm(rating ~ 1, data = wine)
> ll.null <- fm0$logLik
> ## The null or total deviance:
> (Deviance <- -2 * (ll.null - ll.full)) ## 39.407
```

**Example 8:** An ANODEV table for the wine data in Table 1 is presented in Table 5 where the total deviance is broken up into model deviance (due to treatments) and residual deviance.
Table 5: ANODE table for the data in Table 1.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Deviance</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>12</td>
<td>39.407</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Treatment</td>
<td>3</td>
<td>34.606</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Temperature, $T$</td>
<td>1</td>
<td>26.928</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Contact, $C$</td>
<td>1</td>
<td>11.043</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Interaction, $T \times C$</td>
<td>1</td>
<td>0.1514</td>
<td>0.6972</td>
</tr>
<tr>
<td>Residual</td>
<td>9</td>
<td>4.8012</td>
<td>0.8513</td>
</tr>
</tbody>
</table>

Further, the treatment deviance is described by contributions from main effects and interaction. Observe that the deviances for the main effects and interaction do not add up to the deviance for Treatment as the corresponding sums of squares would have in an analogous linear model (ANOVA)\(^9\). The deviances for these terms can instead be interpreted as likelihood ratio tests of nested models: the deviance for the interaction term is the likelihood ratio statistics of the interaction controlling for the main effects, and the deviances for the main effects are the likelihood ratio statistics for these terms while controlling for the other main effect and ignoring the interaction term. As is clear from Table 5, there are significant treatment differences and these seem to describe the data well since the residual deviance is insignificant—the latter is a goodness of fit test for the cumulative logit model describing treatment differences. Further, the treatment differences are well captured by the main effects and there is no indication of an important interaction.

The terminology can be a bit confusing in this area. Sometimes any difference in deviance between two nested models, i.e., a likelihood ratio statistic is denoted a deviance and sometimes any quantity that is proportional to minus twice the log-likelihood of a model is denoted the deviance of that model.

2.5.3 Goodness of fit tests with the deviance

The deviance can be used to test the goodness of fit of a particular reduced model. The deviance asymptotically follows as $\chi^2$ distribution with degrees of freedom equal to the difference in the number of parameters between the two models. The asymptotics are generally good if the expected frequencies under the reduced model are not too small and as a general rule they should all be at least five. This provides a goodness of fit test of the reduced model. The expectation of a random variable that follows a $\chi^2$-distribution is equal to the degrees of freedom of the distribution, so as a rule of thumb, if the deviance in (10) is about the same size as the difference in the number of parameters, there is not evidence of lack of fit.

One problem with the deviance for a particular (reduced) model is that it depends on which model is considered the full model, i.e., how the total deviance is calculated, which often derives from the tabulation of the data. Observe that differences in deviance for nested models are independent of the likelihood of a full model, so deviance differences are insensitive to this choice. Collett (2002) recommends that the data are aggregated as much as possible when evaluating deviances and goodness of fit tests are performed.

**Example 9:** In the presentation of the wine data in example 1 and Table 1, the data were aggregated over judges and bottles. Had we included bottle in the tabulation of the data we

\(^9\)This holds for orthogonal designs including balanced and complete tables like Table 1.
Table 6: Table of the wine data similar to Table 1, but including bottle in the tabulation.

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Contact</th>
<th>Bottle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>cold</td>
<td>no</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cold</td>
<td>no</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cold</td>
<td>yes</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>cold</td>
<td>yes</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>warm</td>
<td>no</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>warm</td>
<td>no</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>warm</td>
<td>yes</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>warm</td>
<td>yes</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

would have arrived at Table 6. A full model for the data in Table 1 has \((5-1)(4-1) = 12\) degrees of freedom while a full model for Table 6 has \((5-1)(8-1) = 28\) degrees of freedom and a different deviance.

If it is decided that bottle is not an important variable, Collett’s recommendation is that we base the residual deviance on a full model defined from Table 1 rather than Table 6.

2.6 Latent variable motivation for cumulative link models

A cumulative link model can be motivated by assuming an underlying continuous latent variable, \(S\) with cumulative distribution function, \(F\). The ordinal response variable, \(Y_i\) is then observed in category \(j\) if \(S_i\) is between the thresholds \(\theta_{j-1}^* < S_i \leq \theta_j^*\) where

\[-\infty \equiv \theta_0^* < \theta_1^* < \ldots < \theta_{J-1}^* < \theta_J^* \equiv \infty\]

divide the real line on which \(S\) lives into \(J+1\) intervals. The situation is illustrated in Fig. 2 where a probit link and \(J = 4\) is adopted. The three thresholds, \(\theta_1, \theta_2, \theta_3\) divide the area under the curve into four parts each of which represent the probability of a response falling in the four response categories. The thresholds are fixed on the scale, but the location of the latent distribution, and therefore also the four areas under the curve, changes with \(x_i\).

A normal linear model for the latent variable is

\[S_i = \alpha + x_i^T \beta^* + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)\] (13)

where \(\{\varepsilon_i\}\) are random disturbances and \(\alpha\) is the intercept, i.e., the mean value of \(S_i\) when \(x_i\) correspond to a reference level for factors and to zero for continuous covariates. Equivalently we could write: \(S_i \sim N(\alpha + x_i^T \beta^*, \sigma^2)\).

The cumulative probability of an observation falling in category \(j\) or below is then:

\[\gamma_{ij} = P(Y_i \leq j) = P(S_i \leq \theta_j^*) = P\left(Z_i \leq \frac{\theta_j^* - \alpha - x_i^T \beta^*}{\sigma}\right) = \Phi\left(\frac{\theta_j^* - \alpha - x_i^T \beta^*}{\sigma}\right)\] (14)

where \(Z_i = (S_i - \alpha - x_i^T \beta^*)/\sigma \sim N(0, 1)\) and \(\Phi\) is the standard normal CDF.

Since the absolute location and scale of the latent variable, \(\alpha\) and \(\sigma\) respectively, are not identifiable from ordinal observations, an identifiable model is

\[\gamma_{ij} = \Phi(\theta_j - x_i^T \beta),\] (15)
with identifiable parameter functions:

\[ \theta_j = (\theta_j^* - \alpha)/\sigma \quad \text{and} \quad \beta = \beta^*/\sigma. \] (16)

Observe how the minus in (15) entered naturally such that a positive \( \beta \) means a shift of the latent distribution in a positive direction.

Model (15) is exactly a cumulative link model with a probit link. Other distributional assumptions for \( S \) correspond to other link functions. In general assuming that the cumulative distribution function of \( S \) is \( F \) corresponds to assuming the link function is \( F^{-1} \), cf. Table 3.

Some expositions of the latent variable motivation for cumulative link models get around the identifiability problem by introducing restrictions on \( \alpha \) and \( \sigma \), usually \( \alpha = 0 \) and \( \sigma = 1 \) are chosen, which leads to the same definition of the threshold and regression parameters that we use here. However, it seems misleading to introduce restrictions on unidentifiable parameters. If observations really arise from a continuous latent variable, \( \alpha \) and \( \sigma \) are real unknown parameters and it makes little sense to restrict them to take certain values. This draws focus from the appropriate relative signal-to-ratio interpretation of the parameters evident from (16).

The standard form of the logistic distribution has mean zero and variance \( \pi^2/3 \). The logistic distribution is symmetric and shows a some resemblance with a normal distribution with the same mean and variance in the central part of the distribution; the tails of the logistic distribution are a little heavier than the tails of the normal distribution. In Fig. 3 the normal and logistic distributions are compared with variance \( \pi^2/3 \). Therefore, to a reasonable approximation, the parameters of logit and probit models are related in the following way:

\[ \theta_j^{\text{probit}} \approx \theta_j^{\text{logit}} / (\pi/\sqrt{3}) \quad \text{and} \quad \beta^{\text{probit}} \approx \beta^{\text{logit}} / (\pi/\sqrt{3}), \] (17)

where \( \pi/\sqrt{3} \approx 1.81 \)

**Example 10:** Considering once again the wine data the coefficients from logit and probit models with additive effects of temperature and contact are

```R
> fm1 <- clm(rating ~ contact + temp, data = wine, link = "logit")
> fm2 <- clm(rating ~ contact + temp, data = wine, link = "probit")
> structure(rbind(coef(fm1),coef(fm2)),
dimnames=list(c("logit", "probit"), names(coef(fm1))))
```

![Figure 2: Illustration of a cumulative link model in terms of the latent distribution.](image.png)
Figure 3: Left: densities. Right: distributions of logistic (solid) and normal (dashed) distributions with mean zero and variance $\pi^2/3$ which corresponds to the standard form for the logistic distribution.

\begin{align*}
&\begin{array}{cccccc}
1|2 & 2|3 & 3|4 & 4|5 & \text{contactyes} & \text{tempwarm} \\
\text{logit} & -1.3443834 & 1.2508088 & 3.466887 & 5.006404 & 1.5277977 & 2.503102 \\
\text{probit} & -0.7732627 & 0.7360215 & 2.044680 & 2.941345 & 0.8677435 & 1.499375 \\
\end{array}
\end{align*}

In comparison the approximate probit estimates using (17) are

\begin{align*}
&\begin{array}{cccccc}
1|2 & 2|3 & 3|4 & 4|5 & \text{contactyes} & \text{tempwarm} \\
-0.7411974 & 0.6896070 & 1.9113949 & 2.7601753 & 0.8423190 & 1.3800325 \\
\end{array}
\end{align*}

These estimates are a great deal closer to the real probit estimates than the unscaled logit estimates. The average difference between the probit and approximate probit estimates being -0.079.

2.6.1 More on parameter interpretation

Observe that the regression parameter in cumulative link models, cf. (16) are signal-to-noise ratios. This means that adding a covariate to a cumulative link model that reduces the residual noise in the corresponding latent model will increase the signal-to-noise ratios. Thus adding a covariate will (often) increase the coefficients of the other covariates in the cumulative link model. This is different from linear models, where (in orthogonal designs) adding a covariate does not alter the value of the other coefficients\footnote{But the same thing happens in other generalized linear models, e.g., binomial and Poisson models, where the variance is determined by the mean.}. Bauer (2009), extending work by Winship and Mare (1984) suggests a way to rescale the coefficients such they are comparable in size during model development. See also Fielding (2004).

\begin{example}
Consider the estimate of $\text{temp}$ in models for the wine data ignoring and controlling for $\text{contact}$, respectively:

\begin{verbatim}
> coef(clm(rating ~ temp, data = wine, link = "probit"))['tempwarm']
\end{verbatim}

\begin{verbatim}
(tempwarm) 1.37229
\end{verbatim}
\end{example}
and observe that the estimate of temp is larger when controlling for contact. In comparison the equivalent estimates in linear models are not affected—here we use the observed scores for illustration:

\[ \text{coef(lm(as.numeric(rating) ~ temp, data = wine))['tempwarm']} \]

\[ \text{tempwarm} \]

\[ 1.166667 \]

\[ \text{coef(lm(as.numeric(rating) ~ contact + temp, data = wine))['tempwarm']} \]

\[ \text{tempwarm} \]

\[ 1.166667 \]

In this case the coefficients are exactly identical, but in designs that are not orthogonal and observed studies with correlated covariates they will only be approximately the same. □

Regardless of how the threshold parameters discretize the scale of the latent variable, the regression parameters \( \beta \) have the same interpretation. Thus \( \beta \) have the same meaning whether the ordinal variable is measured in, say, five or six categories. Further, the nature of the model interpretation will not change if two or more categories are amalgamated, while parameter estimates will, of course, not be completely identical. This means that regression parameter estimates can be compared (to the extent that the noise level is the same) across studies where response scales with a different number of response categories are adopted. In comparison, for linear models used on scores, it is not so simple to just combine two scores, and parameter estimates from different linear models are not directly comparable.

If the latent variable, \( S_i \) is approximated by scores assigned to the response variable, denote this variable \( Y_i^* \), then a linear model for \( Y_i^* \) can provide approximate estimates of \( \beta \) by applying (16) for cumulative probit models\(^{11}\). The quality of the estimates rest on a number of aspects:

- The scores assigned to the ordinal response variable should be structurally equivalent to the thresholds, \( \theta^* \) that generate \( Y_i \) from \( S_i \). In particular, if the (equidistant) numbers 1, \ldots, \( J \) are the scores assigned to the response categories, the thresholds, \( \theta^* \) are also assumed to be equidistant.

- The distribution of \( Y_i^* \) should not deviate too much from a bell-shaped curve; especially there should not be too many observations in the end categories.

- By appeal to the central limit theorem the coarsening of \( S_i \) into \( Y_i^* \) will “average out” such that bias due to coarsening is probably small.

This approximate estimation scheme extends to other latent variable distributions than the normal where linear models are exchanged with the appropriate location-scale models, cf. Table 3.

\(^{11}\)these approximate regression parameters could be used as starting values for an iterative algorithm to find the ML estimates of \( \beta \), but we have not found it worth the trouble in our Newton algorithm.
Example 12: Consider the following linear model for the rating scores of the wine data, cf. Table 1:

$$Y_i^* = \alpha + \beta_1 \text{temp}_i + \beta_2 \text{contact}_i + \varepsilon_i \sim N(0, \sigma^2)$$

The relative parameter estimates, $\tilde{\beta}$ are:

```r
> lm1 <- lm(as.numeric(rating) ~ contact + temp, data = wine)
> sd.lm1 <- summary(lm1)$sigma
> coef(lm1)[-1] / sd.lm1

contact  temp
0.791107 1.384437
```

which should be compared with the estimates from the corresponding cumulative probit model:

```r
> fm1 <- clm(rating ~ contact + temp, data = wine, link = "probit")
> coef(fm1)[-1]

contact  temp
0.8677435 1.4993746
```

The relative estimates from the linear model are a lower than the cumulative probit estimates, which is a consequence of the fact that the assumptions for the linear model are not fulfilled. In particular the distance between the thresholds is not equidistant:

```r
> diff(coef(fm1)[1:4])

2 3 4 5
1.5092842 1.3086590 0.8966645
```

while the distribution is probably sufficiently bell-shaped, cf. Fig 4.

2.7 Structured thresholds

In this section we will motivate and describe structures on the thresholds in cumulative link models. Three options are available in clm using the `threshold` argument: flexible, symmetric and equidistant thresholds. The default option is "flexible", which corresponds to the conventional ordered, but otherwise unstructured thresholds. The "symmetric" option restricts the thresholds to be symmetric while the "equidistant" option restricts the
2.7.1 Symmetric thresholds

The basic cumulative link model assumed that the thresholds are constant for all values of \( x_i^T \beta \), that they are ordered and finite but otherwise without structure. In questionnaire type response scales, the question is often of the form “how much do you agree with statement” with response categories ranging from “completely agree” to “completely disagree” in addition to a number of intermediate categories possibly with appropriate anchoring words. In this situation the response scale is meant to be perceived as being symmetric, thus, for example, the end categories are equally far from the central category/categories. Thus, in the analysis of such data it can be relevant to restrict the thresholds to be symmetric or at least test the hypothesis of symmetric thresholds against the more general alternative requiring only that the thresholds are ordered in the conventional cumulative link model.

An example with six response categories and five thresholds is given in Table 7 where the central threshold, \( \theta_3 \), maps to \( c \) while \( a \) and \( b \) are spacings determining the distance to the remaining thresholds. Symmetric thresholds is a parsimonious alternative since three rather than five parameters are required to determine the thresholds in this case. Naturally at least four response categories, i.e., three thresholds are required for the symmetric thresholds to use less parameters than the general alternative. With an even number of thresholds, we use a parameterization with two central thresholds as shown in Table 8.

**Example 13:** I am missing some good data to use here.

2.7.2 Equidistant thresholds

Ordinal data sometimes arise when the intensity of some perception is rated on an ordinal response scale. An example of such a scale is the ratings of the bitterness of wine described in example 1. In such cases it is natural to hypothesize that the thresholds are equally spaced, or equidistant as we shall denote this structure. Equidistant thresholds use only two parameters and our parameterization can be described by the following mapping:

\[
\theta_j = a + b(j - 1), \quad \text{for} \quad j = 1, \ldots, J - 1
\]

such that \( \theta_1 = a \) is the first threshold and \( b \) denotes the distance between adjacent thresholds.

**Example 14:** In example 3 we fitted a model for the wine data (cf. Table 1) with additive effects of temperature and contact while only restricting the thresholds to be suitably ordered. For convenience this model fit is repeated here:
```r
> fm1 <- clm(rating ~ temp + contact, data=wine)
> summary(fm1)

formula: rating ~ temp + contact
data: wine
link    threshold  nobs logLik    AIC niter max.grad  cond.H
logit flexible  72  -86.49 184.98  6(0) 4.01e-12 2.7e+01

Coefficients:

  Estimate  Std. Error  z value Pr(>|z|)
tempwarm  2.5031      0.5287   4.735  2.19e-06 ***
contactyes 1.5278      0.4766   3.205  0.00135 **

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:

  Estimate  Std. Error  z value
1|2  -1.3444      0.5171  -2.600
2|3   1.2508      0.4379   2.857
3|4   3.4669      0.5978   5.800
4|5   5.0064      0.7309   6.850

The successive distances between the thresholds in this model are

> diff(fm1$alpha)

 2|3   2.595192
3|4   2.216078
4|5   1.539517

so the distance between the thresholds seems to be decreasing. However, the standard errors
of the thresholds are about half the size of the distances, so their position is not that well
determined. A model where the thresholds are restricted to be equally spaced is fitted with

> fm2 <- clm(rating ~ temp + contact, data=wine, threshold="equidistant")
> summary(fm2)

formula: rating ~ temp + contact
data: wine
link    threshold  nobs logLik    AIC niter max.grad  cond.H
logit equidistant 72  -87.86 183.73  5(0) 4.80e-07 3.2e+01

Coefficients:

  Estimate  Std. Error  z value Pr(>|z|)
tempwarm  2.4632      0.5164   4.77  1.84e-06 ***
contactyes 1.5080      0.4712   3.20  0.00137 **

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:

  Estimate  Std. Error  z value
threshold.1  -1.0010      0.3978  -2.517
spacing      2.1229      0.2455   8.646

so here \( \hat{\theta}_1 = \hat{a} = -1.001 \) and \( \hat{b} = 2.123 \) in the parameterization of (18). We can test the
assumption of equidistant thresholds against the flexible alternative with a likelihood ratio test:

---

21
```
> anova(fm1, fm2)

Likelihood ratio tests of cumulative link models:

<table>
<thead>
<tr>
<th></th>
<th>no.par</th>
<th>AIC</th>
<th>logLik</th>
<th>LR.stat</th>
<th>df</th>
<th>Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm2</td>
<td>4</td>
<td>183.73</td>
<td>-87.865</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fm1</td>
<td>6</td>
<td>184.98</td>
<td>-86.492</td>
<td>2.7454</td>
<td>2</td>
<td>0.2534</td>
</tr>
</tbody>
</table>

so the $p$-value is $p = 0.253$ not providing much evidence against equidistant thresholds.

3 Assessing the likelihood and model convergence

Cumulative link models are non-linear models and in general the likelihood function of non-linear models is not guaranteed to be well behaved or even uni-modal. However, as mentioned in section 2.4 Pratt (1981) and Burridge (1981) showed that the log-likelihood function of cumulative link models with the link functions considered in Table 3, except for the cauchit link, is concave. This means that there is a unique global optimum so there is no risk of convergence to a local optimum. There is no such guarantee in more general models.

There are no closed form expressions for the ML estimates of the parameters in cumulative link models. This means that iterative methods have to be used to fit the models. There is no guarantee that the iterative method converges to the optimum. In complicated models the iterative method may terminate at a local optimum or just far from the optimum resulting in inaccurate parameter estimates. We may hope that the optimization process warns about non-convergence, but that can also fail. To be sure the likelihood function is well behaved and that an unequivocal optimum has been reached we have to inspect the likelihood function in a neighborhood around the optimum. There is no such guarantee in more general models.

As special feature for cumulative link models, the threshold parameters are restricted to be ordered and therefore naturally bounded. This may cause the log-likelihood function to be irregular for one or more parameters. This also motivates visualizing the likelihood function in the neighborhood of the optimum. To do this we use the slice function. As the name implies, it extracts a (one-dimensional) slice of the likelihood function.

We use the likelihood slices for two purposes:

1. To visualize and inspect the likelihood function in a neighborhood around the optimum. We choose a rather large neighborhood and look at how well-behaved the likelihood function is, how close to a quadratic function it is, and if there is only one optimum.

2. To verify that the optimization converged and that the parameter estimates are accurately determined. For this we choose a rather narrow neighborhood around the optimum.

The log-likelihood slice is defined as

$$\ell_{\text{slice}}(\alpha_a; y) = \ell(\alpha_a, \hat{\alpha}_{-a}; y) ,$$

(19)
where \( \alpha \) denotes the joint parameter vector. Thus the slice is the log-likelihood function regarded as a function of the \( a \)th parameter while the remaining parameters are fixed at their ML estimates.

The Hessian at the optimum measures the curvature in the log-likelihood function in the parameter space. To scale the range of the parameters at which the likelihood is sliced, we consider the following measures of curvature:

\[
\zeta = \sqrt{\text{diag}(-H)}
\]

Here \( \zeta \) is a \((q + p)\)-vector of curvature units for the \( q \) threshold parameters \( \theta \) and the \( p \) regression parameters \( \beta \). We can understand \( \zeta \) as a vector of standard errors which are not controlled for the dependency of the other parameters. Indeed, if the parameters are completely uncorrelated, \( H \) is diagonal and \( \zeta \) coincide with standard errors, cf. (8), however, this is not possible in cumulative link models.

The range of the parameters at which the log-likelihood is sliced is then a simple multiple \((\lambda)\) of the curvature units in each direction: \( \lambda \zeta \). Working in curvature units is a way to standardize the parameter range even if some parameters are much better determined (much less curvature in the log-likelihood) than others.

A well-behave log-likelihood function will be approximately quadratic in shape in a neighborhood around the optimum. Close enough to the optimum, the log-likelihood function will be virtually indistinguishable from a quadratic approximation if all parameters are identified. For the \( a \)th parameter the quadratic approximation to the log-likelihood function is given by

\[
-\frac{(\hat{\alpha}_a - \alpha_a)^2}{2\zeta_a^2}.
\]

**Example 15:** Consider again a cumulative link model for the wine data. In the following we ask for a slice of the log-likelihood function for each of the parameters and plot these. By setting \( \lambda = 5 \) we ask for the slice in a rather wide neighborhood around the optimum:

```r
> fm1 <- clm(rating ~ temp + contact, data=wine)
> slice.fm1 <- slice(fm1, lambda = 5)
> par(mfrow = c(2, 3))
> plot(slice.fm1)
```

The result is shown in Fig. 5. By default the quadratic approximation is included for reference in the plot.

For this model we see that the log-likelihood function is nicely quadratic for the regression parameters while it is less so for the threshold parameters and particularly bad for the end thresholds. The log-likelihood is relative as indicated by the label on the vertical axis since the values are relative to the maximum log-likelihood value.

From Fig. 5 it seems that the parameter estimates as indicated by the vertical bars are close to the optimum indicating successful model convergence. To investigate more closely we slice the likelihood at a much smaller scale using \( \lambda = 10^{-5} \):

```r
> slice2.fm1 <- slice(fm1, lambda = 1e-5)
> par(mfrow = c(2, 3))
> plot(slice2.fm1)
```

The resulting figure is shown in Fig. 6. Observe that 1) the model has converged, 2) from inspection of the horizontal axis all parameters estimates are correct to at least six decimals, 3) the quadratic approximation is indistinguishable from the log-likelihood at this scale and 4) from the vertical axis the log-likelihood value is determined accurately to at least 12 digits. □
Figure 5: Slices of the log-likelihood function for parameters in a model for the bitterness-of-wine data. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars indicate maximum likelihood estimates.
Unfortunately there is no general way to infer confidence intervals from the likelihood slices—for that we have to use the computationally more intensive profile likelihoods. Compared to the profile likelihoods discussed in section 4, the slice is much less computationally demanding since the likelihood function is only evaluated—not optimized, at a range of parameter values.

## 4 Confidence intervals and profile likelihood

Confidence intervals are convenient for summarizing the uncertainty about estimated parameters. The classical symmetric estimates given by $\hat{\beta} \pm z_{1-\alpha/2} \hat{\text{se}}(\hat{\beta})$ are based on the Wald statistic\(^{12}\), $w(\beta) = (\hat{\beta} - \beta) / \hat{\text{se}}(\hat{\beta})$ and available by `confint(fm1, type = "Wald")`. A similar result could be obtained by `confint.default(fm1)`. However, outside linear models asymmetric confidence intervals often better reflect the uncertainty in the parameter estimates. More accurate, and generally asymmetric, confidence intervals can be obtained by using the likelihood root statistic instead; this relies on the so-called profile likelihood

\(^{12}\)where $z_{1-\alpha/2}$ is the $(1 - \alpha/2)$-quantile of the standard normal CDF.
written here for an arbitrary scalar parameter $\beta_a$:

$$\ell_p(\beta_a; y) = \max_{\theta, \beta - a} \ell(\theta, \beta; y),$$

where $\beta - a$ is the vector of regression parameters without the $a$th one. In words, the profile log-likelihood for $\beta_a$ is given as the full log-likelihood optimized over all parameters but $\beta_a$. To obtain a smooth function, the likelihood is optimized over a range of values of $\beta_a$ around the ML estimate, $\hat{\beta}_a$, further, these points are interpolated by a spline to provide an even smoother function.

The likelihood root statistic (see e.g., Pawitan, 2001; Brazzale et al., 2007) is defined as:

$$r(\beta_a) = \text{sign}(\hat{\beta}_a - \beta_a) \sqrt{-2[\ell(\hat{\theta}, \hat{\beta}; y) - \ell_p(\beta_a; y)]}$$

and just like the Wald statistic its reference distribution is the standard normal. Confidence intervals based on the likelihood root statistic are defined as those values of $\beta_a$ for which $r(\beta_a)$ is in between specified bounds, e.g., $-1.96$ and $1.96$ for 95% confidence intervals. Formally the confidence intervals are defined as

$$CI : \{ \beta_a; |r(\beta_a)| < z_{1 - \alpha / 2} \}.$$

**Example 17:** Consider again a model for the wine data. The profile likelihood confidence intervals are obtained with

```r
> fm1 <- clm(rating ~ temp + contact, data=wine)
> confint(fm1)
```

```
2.5 %  97.5 %
tempwarm 1.5097627  3.595225
contactyes 0.6157925  2.492404
```

where we would have appended `type = "profile"` since this is the default. Confidence intervals with other confidence levels are obtained using the `level` argument. The equivalent confidence intervals based on the Wald statistic are

```r
> confint(fm1, type = "Wald")
```

```
2.5 %  97.5 %
1|2  -2.3578848 -0.330882
2|3   0.3925794  2.109038
3|4   2.2952980  4.638476
4|5   3.5738541  6.438954
tempwarm 1.4669081  3.539296
contactyes 0.5936345  2.461961
```

In this case the Wald bounds are a little too low compared to the profile likelihood confidence bounds. □

Visualization of the likelihood root statistic can be helpful in diagnosing non-linearity in the parameterization of the model. The linear scale is particularly suited for this rather than other scales, such as the quadratic scale at which the log-likelihood lives.

The `plot` method for `profile` objects that can produce a range of different plots on different scales. The `plot` method takes the following arguments:
Figure 7: Likelihood root statistic (solid) and Wald statistic (dashed) for the `tempwarm` parameter in the model for the wine data. Horizontal lines indicate 95% and 99% confidence bounds for both statistics.

```r
function (x, which.par = seq_len(nprofiles), level = c(0.95, 0.99),
  Log = FALSE, relative = TRUE, root = FALSE, fig = TRUE, approx = root,
  n = 1000, ask = prod(par("mfcol")) < length(which.par) && dev.interactive(),
  ..., ylim = NULL)
```

If `root = TRUE` an approximately linear plot of $\beta_a$ versus $-r(\beta_a)$ is produced\(^{13}\). If `approx = TRUE` (the default when `root = TRUE`) the Wald statistic is also included on the square root scale, $-\sqrt{w(\beta_a)}$. This is the tangent line to $-r(\beta_a)$ at $\hat{\beta_a}$ and provides a reference against which to measure curvature in $r(\beta_a)$. Horizontal lines at $\pm 1.96$ and $\pm 2.58$ indicate 95% and 99% confidence intervals.

When `root = FALSE`, the `Log` argument controls whether the likelihood should be plotted on the log scale, and similarly the `relative` argument controls whether the absolute or relative (log-) likelihood should be plotted. At the default settings, the `plot` method produce a plot

**Example 18:** Consider the model from the previous example. The likelihood root statistic can be obtained with

```r
> pr1 <- profile(fm1, which.beta="tempwarm")
> plot(pr1, root=TRUE)
```

The resulting figure is shown in Fig. 7. A slight skewness in the profile likelihood for `tempwarm` translates into curvature in the likelihood root statistic. The Wald and profile likelihood confidence intervals are given as intersections with the horizontal lines.

In summarizing the results of a models fit, I find the relative likelihood scale, $\exp(-r(\beta_a)^2/2) = \exp\{\ell(\hat{\theta}, \hat{\beta}; y) - \ell_p(\beta_a; y)\}$ informative. The evidence about the parameters is directly visible on this scale (cf. Fig. 8); the ML estimate has maximum support, and values away from here are less supported by the data, 95% and 99% confidence intervals are readily read of

\(^{13}\)actually we reversed the sign of the statistic in the display since a line from lower-left to upper-right looks better than a line from upper-left to lower-right.
Figure 8: Relative profile likelihoods for the regression parameters in the Wine study. Horizontal lines indicate 95% and 99% confidence bounds.

the plots as intersections with the horizontal lines. Most importantly the plots emphasize that a range of parameter values are actually quite well supported by the data—something which is easy to forget when focus is on the exact numbers representing the ML estimates.

Example 19: The relative profile likelihoods are obtained with

\[
> \text{pr1} \leftarrow \text{profile(fm1, alpha=1e-4)}
\]

\[
> \text{plot(pr1)}
\]

and provided in Fig. 8. From the relative profile likelihood for \texttt{tempwarm} we see that parameter values between 1 and 4 are reasonably well supported by the data, and values outside this range has little likelihood. Values between 2 and 3 are very well supported by the data and all have high likelihood.

\[\square\]

A An approximate ML estimation method for CLMs

The approach with multiple ordinary logistic regression models considered in example 2 and summarized in Table 2 can be improved on by constructing a single ordinary logistic regression model that estimates the regression parameters only once while also estimating all the threshold parameters. This approach estimates the same parameters as the cumulative logit model, but it does not yield ML estimates of the parameters although generally the estimates are quite close. A larger difference is typically seen in the standard errors which are generally too small. This approach is further described by Winship and Mare (1984).

The basic idea is to form a binary response: \(y^* = [I(y \leq 1), I(y \leq 2), \ldots, I(y \leq J - 1)]\) of length \(n(J - 1)\), where originally we had \(n\) observations falling in \(J\) categories. The original design matrix, \(X\) is stacked \(J - 1\) times and indicator variables for \(j = 1, \ldots, J - 1\) are included to give estimates for the thresholds.

Example 20: We now continue example 2 by estimating the parameters with the approach described above. First we form the data set to which we will fit the model:

\[
> \text{tab} \leftarrow \text{with(wine, table(contact, rating))}
\]

\[
> \text{dat} \leftarrow \text{data.frame(freq =c(tab),}
\]

\[
\quad \quad \text{contact=rep(c("no", "yes"), 5),}
\]

\[
\quad \quad \text{rating = factor(rep(1:5, each=2), ordered=TRUE))}
\]

\[
> \text{dat}
\]

\[\text{28}\]
freq contact rating
1 4 no 1
2 1 yes 1
3 14 no 2
4 8 yes 2
5 13 no 3
6 13 yes 3
7 3 no 4
8 9 yes 4
9 2 no 5
10 5 yes 5

The cumulative link model would be fitted with

```r
> fm1 <- clm(rating ~ contact, weights=freq)
```

Then we generate the new response and new data set:

```r
> thresholds <- 1:4
> cum.rate <- as.vector(sapply(thresholds, function(x) dat$rating <= x))
> rating.factor <- gl(n=length(thresholds), k=nrow(dat),
                     length=nrow(dat) * length(thresholds))
> thres.X <- model.matrix(~ rating.factor - 1)
> old.X <- -model.matrix(~contact, dat)[, -1, drop=FALSE]
> new.X <- kronecker(matrix(rep(1, length(thresholds)), nc = 1), old.X)
> weights <- kronecker(matrix(rep(1, length(thresholds)), nc = 1), dat$freq)
> new.X <- cbind(thres.X, new.X)
> colnames(new.X)[-seq(length(thresholds))] <- colnames(old.X)
> p.df <- cbind(cum.rate = 1*cum.rate, as.data.frame(new.X), weights)
> p.df
```

<table>
<thead>
<tr>
<th>cum.rate</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>contactyes</th>
<th>weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>13</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>9</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>5</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>8</td>
</tr>
</tbody>
</table>

29
where the first column is the new binary response variable, the next four columns are indicator variables for the thresholds, the following column is the indicator variable for contact and the last column holds the weights. Observe that while the original data set had 10 rows, the new data set has 40 rows. We fit the ordinary logistic regression model for these data with

```r
> glm1 <- glm(cum.rate ~ t1+t2 +t3 +t4 - 1 + contactyes,
weights=weights, family=binomial, data=p.df)
> summary(glm1)
```

**Coefficients:**

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|----------|
| t1       | -2.14176   | 0.47616 | -4.498   | 6.86e-06 *** |
| t2       | 0.04797    | 0.29048 | 0.165    | 0.86883     |
| t3       | 1.71907    | 0.35132 | 4.893    | 9.93e-07 *** |
| t4       | 2.97786    | 0.47203 | 6.309    | 2.81e-10 *** |
| contactyes | 1.21160   | 0.33593 | 3.607    | 0.00031 *** |

**Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 399.25 on 40 degrees of freedom
Residual deviance: 246.47 on 35 degrees of freedom
AIC: 256.47

Number of Fisher Scoring iterations: 5

Notice that we suppressed the intercept since the thresholds play the role as intercepts. In comparing these estimates with the estimates from the genuine cumulative logit model presented in the last two columns of Table 2 we see that these approximate ML estimates are remarkably close to the true ML estimates. Observe however that the standard error of the contact effect
is much smaller; this is due to the fact that the model was fitted to \( n(J - 1) \) observations while originally we only had \( n \) of those. The weights could possibly be modified to adjust for this, but we will not pursue this any further.

In lack of efficient software that yields ML estimates of the cumulative logit model, this approach could possibly be justified. This might have been the case some 20 years ago. However, in the present situation this approximate approach is only cumbersome for practical applications. □

References


Appendix K

A Tutorial on fitting Cumulative Link Models with the ordinal Package

A Tutorial on fitting Cumulative Link Models with the \texttt{ordinal} Package

Rune Haubo B Christensen

September 10, 2012

Abstract

It is shown by example how a cumulative link mixed model is fitted with the \texttt{clm} function in package \texttt{ordinal}. Model interpretation and inference is briefly discussed.

Contents

1 Introduction 2
2 Fitting Cumulative Link Models 3
3 Nominal Effects 6
4 Scale Effects 8
5 Structured Thresholds 9
6 Predictions 11
7 Infinite Parameter Estimates 11
8 Unidentified parameters 12
9 Assessment of Model Convergence 14
10 Profile Likelihood 16
Table 1: Ratings of the bitterness of some white wines. Data are adopted from Randall (1989).

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Contact</th>
<th>Bottle</th>
<th>Judge</th>
</tr>
</thead>
<tbody>
<tr>
<td>cold no</td>
<td>1</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>cold yes</td>
<td>2</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>cold yes</td>
<td>3</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>cold yes</td>
<td>4</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>warm no</td>
<td>5</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>warm yes</td>
<td>6</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>warm yes</td>
<td>7</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>warm yes</td>
<td>8</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
</tbody>
</table>

1 Introduction

We will consider the data on the bitterness of wine from Randall (1989) presented in Table 1 and available as the object `wine` in package `ordinal`. The data were also analyzed with mixed effects models by Tutz and Hennevogl (1996). The following gives an impression of the wine data object:

```r
> data(wine)
> head(wine)

response rating temp contact bottle judge
1 36 2 cold no 1 1
2 48 3 cold no 2 1
3 47 3 cold yes 3 1
4 67 4 cold yes 4 1
5 77 4 warm no 5 1
6 60 4 warm no 6 1
```

The data represent a factorial experiment on factors determining the bitterness of wine with 1 = “least bitter” and 5 = “most bitter”. Two treatment factors (temperature and contact) each have two levels. Temperature and contact between juice and skins can be controlled when crushing grapes during wine production. Nine judges each assessed wine from two bottles from each of the four treatment conditions, hence there are 72 observations in all. For more information see the manual entry for the wine data: `help(wine)`.

The intention with this tutorial is not to explain cumulative link models in details, rather the main aim is to briefly cover the main functions and methods in the `ordinal` package to analyze ordinal data. A more thorough introduction to cumulative link models and the `ordinal` package is given in Christensen (2011); a book length treatment of ordinal data...
analysis is given by Agresti (2010) although not related to the ordinal package.

2 Fitting Cumulative Link Models

We will fit the following cumulative link model to the wine data:

\[
\text{logit}(P(Y_i \leq j)) = \theta_j - \beta_1(\text{temp}_i) - \beta_2(\text{contact}_i) \quad i = 1, \ldots, n, \quad j = 1, \ldots, J - 1
\]  

(1)

This is a model for the cumulative probability of the \(i\)th rating falling in the \(j\)th category or below, where \(i\) index all observations \((n = 72)\) and \(j = 1, \ldots, J\) index the response categories \((J = 5)\). The \(\theta_j\) parameter is the intercept for the \(j\)th cumulative logit \(\text{logit}(P(Y_i \leq j))\); they are known as threshold parameters, intercepts or cut-points.

This model is also known as the proportional odds model, a cumulative logit model, and an ordered logit model.

We fit this cumulative link model by maximum likelihood with the clm function in package ordinal. Here we save the fitted clm model in the object \(fm1\) (short for fitted model 1) and print the model by simply typing its name:

```r
> fm1 <- clm(rating ~ temp + contact, data=wine)
> fm1
```

```
formula: rating ~ temp + contact
data: wine

link threshold nobs logLik AIC niter max.grad
logit flexible 72 -86.49 184.98 6(0) 4.01e-12

Coefficients:
  tempwarm contactyes
  2.5031 1.5278

Threshold coefficients:
  1|2 2|3 3|4 4|5
-1.344 1.251 3.467 5.006

Additional information is provided with the summary method:
> summary(fm1)
```

```
formula: rating ~ temp + contact
data: wine

logLik AIC niter max.grad cond.H
logit flexible 72 -86.49 184.98 6(0) 4.01e-12 2.7e+01

Coefficients:

|            | Estimate | Std. Error | z value | Pr(>|z|) |
|------------|----------|------------|---------|----------|
| tempwarm   | 2.5031   | 0.5287     | 4.735   | 2.19e-06 *** |
| contactyes | 1.5278   | 0.4766     | 3.205   | 0.00135 **  |

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 1
Threshold coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-1.3444</td>
<td>0.5171</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1.2508</td>
<td>0.4379</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3.4669</td>
<td>0.5978</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5.0064</td>
<td>0.7309</td>
</tr>
</tbody>
</table>

The primary result is the coefficient table with parameter estimates, standard errors and Wald (or normal) based p-values for tests of the parameters being zero. The maximum likelihood estimates of the parameters are:

\[ \hat{\beta}_1 = 2.50, \quad \hat{\beta}_2 = 1.53, \quad \{\hat{\theta}_j\} = \{-1.34, 1.25, 3.47, 5.01\} \]  

(2)

The number of Newton-Raphson iterations is given below niter with the number of step-halvings in parenthesis. max.grad is the maximum absolute gradient of the log-likelihood function with respect to the parameters. A small absolute gradient is a necessary condition for convergence of the model. The iterative procedure will declare convergence whenever the maximum absolute gradient is below

> clm.control()$gradTol

[1] 1e-06

which may be altered — see help(clm.control).

The condition number of the Hessian (cond.H) measures the empirical identifiability of the model. High numbers, say larger than $10^4$ or $10^6$ indicate that the model is ill defined. This could indicate that the model can be simplified, that possibly some parameters are not identifiable, and that optimization of the model can be difficult. In this case the condition number of the Hessian does not indicate a problem with the model.

The coefficients for temp and contact are positive indicating that higher temperature and more contact increase the bitterness of wine, i.e., rating in higher categories is more likely. The odds ratio of the event \( Y \geq j \) is \( \exp(\beta_{\text{treatment}}) \), thus the odds ratio of bitterness being rated in category \( j \) or above at warm relative to cold temperatures is

> exp(coef(fm1)[5])

tempwarm

12.22034

The p-values for the location coefficients provided by the summary method are based on the so-called Wald statistic. More accurate tests are provided by likelihood ratio tests. These can be obtained with the anova method, for example, the likelihood ratio test of contact is

> fm2 <- clm(rating ~ temp, data=wine)
> anova(fm2, fm1)

Likelihood ratio tests of cumulative link models:

<table>
<thead>
<tr>
<th></th>
<th>no.par</th>
<th>AIC</th>
<th>logLik</th>
<th>LR.stat</th>
<th>df</th>
<th>Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm2</td>
<td>5</td>
<td>194.03</td>
<td>-92.013</td>
<td>-</td>
<td>92.013</td>
<td></td>
</tr>
</tbody>
</table>
which in this case produces a slightly lower p-value. Equivalently we can use `drop1` to obtain likelihood ratio tests of the explanatory variables while controlling for the remaining variables:

```
> drop1(fm1, test = "Chi")
```

Single term deletions

Model:

```
rating ~ temp + contact
```

<table>
<thead>
<tr>
<th>Df</th>
<th>AIC</th>
<th>LRT</th>
<th>Pr(&gt;Chi)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>184.98</td>
<td>26.928</td>
<td>2.112e-07 ***</td>
</tr>
<tr>
<td>temp</td>
<td>1 209.91</td>
<td>11.043</td>
<td>0.0008902 ***</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' ...

Likelihood ratio tests of the explanatory variables while ignoring the remaining variables are provided by the `add1` method:

```
> fm0 <- clm(rating ~ 1, data=wine)
> add1(fm0, scope = ~ temp + contact, test = "Chi")
```

Single term additions

Model:

```
rating ~ 1
```

<table>
<thead>
<tr>
<th>Df</th>
<th>AIC</th>
<th>LRT</th>
<th>Pr(&gt;Chi)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>215.44</td>
<td>23.411</td>
<td>1.308e-06 ***</td>
</tr>
<tr>
<td>temp</td>
<td>1 194.03</td>
<td>7.5263</td>
<td>0.00608 **</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' ...

In this case these latter tests are not as strong as the tests controlling for the other variable.

Confidence intervals are provided by the `confint` method:

```
> confint(fm1)
```

<table>
<thead>
<tr>
<th></th>
<th>2.5 %</th>
<th>97.5 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>tempwarm</td>
<td>1.5097627</td>
<td>3.595225</td>
</tr>
<tr>
<td>contactyes</td>
<td>0.6157925</td>
<td>2.492404</td>
</tr>
</tbody>
</table>

These are based on the profile likelihood function and generally fairly accurate. Less accurate, but simple and symmetric confidence intervals based on the standard errors of the parameters (so-called Wald confidence intervals) can be obtained with

```
> confint(fm1, type="Wald")
```

<table>
<thead>
<tr>
<th></th>
<th>2.5 %</th>
<th>97.5 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-2.3578848</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.3925974</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2.2952980</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3.5738541</td>
</tr>
<tr>
<td>tempwarm</td>
<td>1.4669081</td>
<td>3.539296</td>
</tr>
</tbody>
</table>
In addition to the logit link, which is the default, the probit, log-log, complementary log-log and cauchit links are also available. For instance, a proportional hazards model for grouped survival times is fitted using the complementary log-log link:

```r
> fm.cll <- clm(rating ~ contact + temp, data=wine, link="cloglog")
```

The cumulative link model in (1) assume that the thresholds, \( \{ \theta_j \} \) are constant for all values of the remaining explanatory variables, here `temp` and `contact`. This is generally referred to as the proportional odds assumption or equal slopes assumption. We can relax that assumption in two general ways: with nominal effects and scale effects which we will now discuss in turn.

### 3 Nominal Effects

The CLM in (1) specifies a structure in which the regression parameters, \( \beta \) are not allowed to vary with \( j \). Nominal effects relax this assumption by allowing one or more regression parameters to vary with \( j \). In the following model we allow the regression parameter for `contact` to vary with \( j \):

\[
\logit(P(Y_i \leq j)) = \theta_j - \beta_1(temp_i) - \beta_2(contact_i)
\]

This means that there is one estimate of \( \beta_2 \) for each \( j \). This model is specified as follows with `clm`:

```r
> fm.nom <- clm(rating ~ temp, nominal=~contact, data=wine)
> summary(fm.nom)
```

```
formula: rating ~ temp
nominal: ~contact
data: wine

link threshold nobs logLik AIC niter max.grad cond.H
logit flexible 72 -86.21 190.42 6(0) 1.64e-10 4.8e+01

Coefficients:
Estimate Std. Error z value Pr(>|z|)
tempwarm 2.519 0.535 4.708 2.5e-06 ***

---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.3230</td>
<td>-1.2464</td>
<td>3.5500</td>
<td>4.6602</td>
<td>-1.6151</td>
<td>-1.5116</td>
<td>-1.6748</td>
<td>-1.0506</td>
<td>1.1618</td>
<td>-1.3900</td>
<td>0.7623</td>
<td>0.4748</td>
<td>0.6560</td>
<td>0.8604</td>
<td>1.1618</td>
<td>-1.3900</td>
</tr>
<tr>
<td>0.5623</td>
<td>0.4748</td>
<td>0.6560</td>
<td>0.8604</td>
<td>-2.3530</td>
<td>-2.625</td>
<td>-5.4110</td>
<td>-5.4162</td>
<td>1.1618</td>
<td>-1.3900</td>
<td>0.7623</td>
<td>0.4748</td>
<td>0.6560</td>
<td>0.8604</td>
<td>1.1618</td>
<td>-1.3900</td>
</tr>
<tr>
<td>-2.3530</td>
<td>-2.625</td>
<td>-5.4110</td>
<td>-5.4162</td>
<td>0.0562</td>
<td>-2.559</td>
<td>-2.581</td>
<td>-1.1720</td>
<td>0.0562</td>
<td>-2.559</td>
<td>-2.581</td>
<td>-1.1720</td>
<td>0.0562</td>
<td>-2.559</td>
<td>-2.581</td>
<td>-1.1720</td>
</tr>
</tbody>
</table>
```

---

6
As can be seen from the output of `summary` there is no regression coefficient estimated for `contact`, but there are two sets of threshold parameters estimated.

The first five threshold parameters have .(Intercept) appended their names indicating that these are the estimates of $\theta_j$. The following five threshold parameters have .contactyes appended their name indicating that these parameters are differences between the threshold parameters at the two levels of contact. This interpretation corresponds to the default treatment contrasts; if other types of contrasts are specified, the interpretation is a little different. As can be seen from the output, the effect of `contact` is almost constant across thresholds and around 1.5 corresponding to the estimate from `fm1` on page 4, so probably there is not much evidence that the effect of `contact` varies with $j$.

We can perform a likelihood ratio test of the equal slopes or proportional odds assumption for `contact` by comparing the likelihoods of models (1) and (3) as follows:

```r
> anova(fm1, fm.nom)
Likelihood ratio tests of cumulative link models:

formula: nominal: link: threshold:
 fm1 rating ~ temp + contact ~1 logit flexible
 fm.nom rating ~ temp "contact logit flexible

no.par AIC logLik LR.stat df Pr(>Chisq)
 fm1 6 184.98 -86.492
 fm.nom 9 190.42 -86.209 0.5667 3 0.904

There is only little difference in the log-likelihoods of the two models, so the test is insignificant. There is therefore no evidence that the proportional odds assumption is violated for `contact`.

It is not possible to estimate both $\beta_2$ and $\beta_{2j}$ in the same model. Consequently variables that appear in `nominal` cannot enter elsewhere as well. For instance not all parameters are identifiable in the following model:

```r
> fm.nom2 <- clm(rating ~ temp + contact, nominal="contact, data=wine)
```

We are made aware of this when summarizing or printing the model:

```r
> summary(fm.nom2)

formula: rating ~ temp + contact
nominal: "contact

Coefficients:

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|----------|
| tempwarm | 2.519      | 0.535   | 4.708    | 2.5e-06 *** |
| contactyes | NA | NA | NA   | 1

Threshold coefficients:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.(Intercept)</td>
<td>-1.3230</td>
</tr>
</tbody>
</table>
```

1
4 Scale Effects

Scale effects are usually motivated from the latent variable interpretation of a CLM. Assume the following model for a latent variable:

\[ S_i = \alpha^* + x_i^T \beta^* + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^*^2) \] (4)

If the ordinal variable \( Y_i \) is observed such that \( Y_i = j \) is recorded if \( \theta^*_{j-1} < S_i \leq \theta^*_j \), where

\[-\infty \equiv \theta^*_0 < \theta^*_1 < \ldots < \theta^*_{j-1} < \theta^*_j \equiv \infty \] (5)

then we have the cumulative link model for \( Y_i \):

\[ P(Y_i \leq j) = \Phi(\theta_j - x_i^T \beta) \]

where we have used \( \theta_j = (\theta^*_j + \alpha^*)/\sigma^* \) and \( \beta = \beta^*/\sigma^* \) (parameters with a “∗” exist on the latent scale, while those without are identifiable), and \( \Phi \) is the inverse probit link and denotes the standard normal CDF. Other assumptions on the distribution of the latent variable, \( S_i \) lead to other link functions, e.g., a logistic distribution for the latent variable corresponds to a logit link.

If the scale (or dispersion) of the latent distribution is described by a log-linear model such that

\[ \log(\sigma_i) = z_i^T \zeta \] (equivalently \( \sigma_i = \exp(z_i^T \zeta) \); also note that \( z_i^T \zeta \) is a linear model just like \( x_i^T \beta \)), then the resulting CLM reads (for more details, see e.g., Christensen (2011) or Agresti (2010)):

\[ P(Y_i \leq j) = \Phi \left( \frac{\theta_j - x_i^T \beta}{\sigma_i} \right) \] (6)

The conventional link functions in cumulative link models correspond to distributions for the latent variable that are members of the location-scale family of distributions (cf. http://en.wikipedia.org/wiki/Location-scale_family). They have the common form \( F(\mu, \sigma) \), where \( \mu \) denotes the location of the distribution and \( \sigma \) denotes the scale of the distribution. For instance in the normal distribution (probit link) \( \mu \) is the mean, and \( \sigma \) is the spread, while in the logistic distribution (logit link), \( \mu \) is the mean and \( \sigma \pi/\sqrt{3} \) is the spread (cf. http://en.wikipedia.org/wiki/Logistic_distribution).

Thus allowing for scale effects corresponds to modelling not only the location of the latent distribution, but also the scale. Just as the absolute location (\( \alpha^* \)) is not identifiable, the absolute scale (\( \sigma^* \)) is not identifiable either in the CLM, however differences in location modelled with \( x_i^T \beta \) and ratios of scale modelled with \( \exp(z_i^T \zeta) \) are identifiable.

Now we turn to our running example and fit a model where we allow the scale of the latent distribution to depend on temperature:

\[ \logit(P(Y_i \leq j)) = \theta_j - \beta_1(\text{temp}_i) - \beta_2(\text{contact}_i) \exp(\zeta_1(\text{temp}_i)) \] (7)
\[ i = 1, \ldots, n, \quad j = 1, \ldots, J - 1 \]

We can estimate this model with

\[
\text{> fm.sca} \leftarrow \text{clm(rating } \sim \text{ temp + contact, scale=} \sim \text{temp, data=wine)}
\]

\[
\text{> summary(fm.sca)}
\]

formula: rating \( \sim \) temp + contact
scale: \( \sim \) temp
data: wine

<table>
<thead>
<tr>
<th>link</th>
<th>threshold</th>
<th>nobs</th>
<th>logLik</th>
<th>AIC</th>
<th>niter</th>
<th>max.grad</th>
<th>cond.H</th>
</tr>
</thead>
<tbody>
<tr>
<td>logit</td>
<td>flexible</td>
<td>72</td>
<td>-86.44</td>
<td>186.88</td>
<td>8(0)</td>
<td>5.25e-09</td>
<td>1.0e+02</td>
</tr>
</tbody>
</table>

Coefficients:

\[
\begin{align*}
\text{Estimate} & \quad \text{Std. Error} & \quad \text{z value} & \quad \text{Pr(>|z|)} \\
\text{tempwarm} & \quad 2.6294 & \quad 0.6860 & \quad 3.833 & \quad 0.000127^{***} \\
\text{contactyes} & \quad 1.5878 & \quad 0.5301 & \quad 2.995 & \quad 0.002743^{**} \\
\end{align*}
\]

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

log-scale coefficients:

\[
\begin{align*}
\text{Estimate} & \quad \text{Std. Error} & \quad \text{z value} & \quad \text{Pr(>|z|)} \\
\text{tempwarm} & \quad 0.09536 & \quad 0.29414 & \quad 0.324 & \quad 0.746 \\
\end{align*}
\]

Threshold coefficients:

\[
\begin{align*}
\text{Estimate} & \quad \text{Std. Error} & \quad \text{z value} & \quad \text{Pr(>|z|)} \\
1|2 & \quad -1.3520 & \quad 0.5223 & \quad -2.588 \\
2|3 & \quad 1.2730 & \quad 0.4533 & \quad 2.808 \\
3|4 & \quad 3.6170 & \quad 0.7774 & \quad 4.653 \\
4|5 & \quad 5.2982 & \quad 1.2027 & \quad 4.405 \\
\end{align*}
\]

Notice that both location and scale effects of temp are identifiable. Also notice that the scale coefficient for temp is given on the log-scale, where the Wald test is more appropriate.

The absolute scale of the latent distribution is not estimable, but we can estimate the scale at warm conditions relative to cold conditions. Therefore the estimate of \( \kappa \) in the relation

\[
\sigma_{\text{warm}} = \kappa \sigma_{\text{cold}}
\]

is given by

\[
\text{> exp(fm.sca}\$\text{zeta)}
\]

tempwarm

1.100054

However, the scale difference is not significant in this case as judged by the \( p \)-value in the summary output. confint and anova apply with no change to models with scale, but drop1, add1 and step methods will only drop or add terms to the (location) formula and not to scale.

5 Structured Thresholds

In section 3 we relaxed the assumption that regression parameters have the same effect across all thresholds. In this section we will instead impose additional restrictions on the thresholds. In the following model we require that the thresholds, \( \theta_j \) are equidistant or equally spaced (\( \theta_j - \theta_{j-1} = \text{constant for } j = 2, \ldots, J - 1 \)):
```r
> fm.equi <- clm(rating ~ temp + contact, data=wine,
thresho"d="equidistant")
> summary(fm.equi)

formula: rating ~ temp + contact
data: wine

link threshold nobs logLik AIC niter max.grad cond.H
logit equidistant 72 -87.86 183.73 5(0) 4.80e-07 3.2e+01

Coefficients:

|          | Estimate | Std. Error | z value | Pr(>|z|) |
|----------|----------|------------|---------|----------|
| tempwarm | 2.4632   | 0.5164     | 4.77    | 1.84e-06 *** |
| contactyes | 1.5080  | 0.4712     | 3.20    | 0.00137 **  |

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold.1</td>
<td>-1.0010</td>
<td>0.3978</td>
<td>-2.517</td>
</tr>
<tr>
<td>spacing</td>
<td>2.1229</td>
<td>0.2455</td>
<td>8.646</td>
</tr>
</tbody>
</table>

The parameters determining the thresholds are now the first threshold and the spacing among consecutive thresholds. The mapping to this parameterization is stored in the transpose of the Jacobian (tJac) component of the model fit. This makes it possible to get the thresholds imposed by the equidistance structure with

```r
c(with(fm.equi, tJac %*% alpha))
```

[1] -1.001044 1.121892 3.244828 5.367764

The following shows that the distance between consecutive thresholds in fm1 is very close to the spacing parameter from fm.equi:

```r
> mean(diff(fm1$alpha))
```

[1] 2.116929

The gain in imposing additional restrictions on the thresholds is the use of fewer parameters. Whether the restrictions are warranted by the data can be tested in a likelihood ratio test:

```r
> anova(fm1, fm.equi)
```

Likelihood ratio tests of cumulative link models:

<table>
<thead>
<tr>
<th></th>
<th>no. par</th>
<th>AIC</th>
<th>logLik</th>
<th>LR.stat</th>
<th>df</th>
<th>Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm.equi</td>
<td>4</td>
<td>183.73</td>
<td>-87.865</td>
<td></td>
<td>2</td>
<td>0.2534</td>
</tr>
<tr>
<td>fm1</td>
<td>6</td>
<td>184.98</td>
<td>-86.492</td>
<td>2.7454</td>
<td>2</td>
<td>0.2534</td>
</tr>
</tbody>
</table>

In this case the test is non-significant, so there is no considerable loss of fit at the gain of saving two parameters, hence we may retain the model with equally spaced thresholds.
```
6 Predictions

Fitted values are extracted with e.g., `fitted(fm1)` and produce fitted probabilities, i.e., the ith fitted probability is the probability that the ith observation falls in the response category that it did. The predictions of which response class the observations would be most likely to fall in given the model are obtained with:

```r
> predict(fm1, type = "class")
```

$fit

```
[1] 2 2 3 3 3 4 4 2 2 3 3 3 4 4 2 2 3 3 3 4 4 2 2 3 3 3 4 4 2 2 3 3 3 4 4 2 2 3 3 3 4 4 2 2 3 3 3 4 4
```

Levels: 1 2 3 4 5

Say we just wanted the predictions for the four combinations of `temp` and `contact`. The probability that an observation falls in each of the five response categories based on the fitted model is given by:

```r
> newData <- expand.grid(temp=levels(wine$temp), contact=levels(wine$contact))
> cbind(newData, predict(fm1, newdata=newData)$fit)
```

<table>
<thead>
<tr>
<th>temp</th>
<th>contact</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>cold</td>
<td>no</td>
<td>0.206790132</td>
<td>0.57064970</td>
<td>0.1922909</td>
<td>0.02361882</td>
<td>0.00665041</td>
</tr>
<tr>
<td>warm</td>
<td>no</td>
<td>0.020887709</td>
<td>0.20141572</td>
<td>0.5015755</td>
<td>0.20049402</td>
<td>0.07562701</td>
</tr>
<tr>
<td>cold</td>
<td>yes</td>
<td>0.053546010</td>
<td>0.37764614</td>
<td>0.4430599</td>
<td>0.09582084</td>
<td>0.02992711</td>
</tr>
<tr>
<td>warm</td>
<td>yes</td>
<td>0.004608274</td>
<td>0.05380128</td>
<td>0.3042099</td>
<td>0.36359581</td>
<td>0.27378469</td>
</tr>
</tbody>
</table>

Standard errors and confidence intervals of predictions are also available, for example, the predictions, standard errors and 95% confidence intervals are given by (illustrated here using `do.call` for the first six observations):

```r
> head(do.call("cbind", predict(fm1, se.fit=TRUE, interval=TRUE)))
```

```
    fit se.fit   lwr  upr
[1,] 0.57064970 0.08683884 0.39887109 0.7269447
[2,] 0.19229094 0.06388672 0.09609419 0.3477399
[3,] 0.44305990 0.07939754 0.29746543 0.5991420
[4,] 0.09582084 0.04257593 0.03887676 0.2173139
[5,] 0.20049402 0.06761012 0.09886604 0.3643505
[6,] 0.20049402 0.06761012 0.09886604 0.3643505
```

The confidence level can be set with the `level` argument and other types of predictions are available with the `type` argument.

7 Infinite Parameter Estimates

If we attempt to test the proportional odds assumption for `temp`, some peculiarities show up:

```r
> fm.nom2 <- clm(rating ~ contact, nominal=~temp, data=wine)
> summary(fm.nom2)
```

```
formula: rating ~ contact
nominal: ~temp
data: wine
11
```

```
```
Several of the threshold coefficients are extremely large with huge standard errors. Also the condition number of the Hessian is very large and a larger number of iterations was used all indicating that something is not well-behaved. The problem is that the the ML estimates of some of the threshold parameters are at (plus/minus) infinity. \texttt{clm} is not able to detect this and just stops iterating when the likelihood function is flat enough for the termination criterion to be satisfied, i.e., when the maximum absolute gradient is small enough.

Even though some parameter estimates are not at (plus/minus) infinity while they should have been, the remaining parameters are accurately determined and the value of the loglikelihood is also accurately determined. This means that likelihood ratio tests are still available, for example, it is still possible to test the proportional odds assumption for \texttt{temp}:

\begin{verbatim}
> anova(fm1, fm.nom2)
Likelihood ratio tests of cumulative link models:

formula: nominal: link: threshold:
fm1 rating ~ temp + contact ~1 logit flexible
fm.nom2 rating ~ contact ~temp logit flexible

no.par AIC logLik LR.stat df  Pr(>Chisq)
fm1 6 184.98 -86.492
fm.nom2 9 187.81 -84.904 3.175 3 0.3654
\end{verbatim}

### 8 Unidentified parameters

In the following example (using another data set) one parameter is not identifiable:

\begin{verbatim}
> data(soup)
> fm.soup <- clm(SURENESS ~ PRODID * DAY, data=soup)
> summary(fm.soup)
\end{verbatim}
formula: SURENESS ~ PRODID * DAY
data: soup

link threshold nobs logLik AIC niter max.grad cond.H
logit flexible 1847 -2672.08 5374.16 6(1) 5.55e-13 9.4e+02

Coefficients: (1 not defined because of singularities)

|                  | Estimate | Std. Error | z value | Pr(>|z|) |
|------------------|----------|------------|---------|----------|
| PRODID2          | 0.6665   | 0.2146     | 3.106   | 0.00189  **|
| PRODID3          | 1.2418   | 0.1784     | 6.959   | 3.42e-12 ***|
| PRODID4          | 0.6678   | 0.2197     | 3.040   | 0.00237  **|
| PRODID5          | 1.1194   | 0.2400     | 4.663   | 3.11e-06 ***|
| PRODID6          | 1.3503   | 0.2337     | 5.779   | 7.53e-09 ***|
| DAY2             | -0.4134  | 0.1298     | -3.186  | 0.00144  **|
| PRODID2:DAY2     | 0.4390   | 0.2590     | 1.695   | 0.09006  .|
| PRODID3:DAY2     | NA       | NA         | NA      | NA       |
| PRODID4:DAY2     | 0.3308   | 0.3056     | 1.083   | 0.27892  |
| PRODID5:DAY2     | 0.3871   | 0.3248     | 1.192   | 0.23329  |
| PRODID6:DAY2     | 0.5067   | 0.3350     | 1.513   | 0.13030  |

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>z value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>-1.63086</td>
<td>0.10740</td>
<td>-15.184</td>
</tr>
<tr>
<td>2/3</td>
<td>-0.64177</td>
<td>0.09682</td>
<td>-6.629</td>
</tr>
<tr>
<td>3/4</td>
<td>-0.31295</td>
<td>0.09546</td>
<td>-3.278</td>
</tr>
<tr>
<td>4/5</td>
<td>-0.05644</td>
<td>0.09508</td>
<td>-0.594</td>
</tr>
<tr>
<td>5/6</td>
<td>0.61692</td>
<td>0.09640</td>
<td>6.399</td>
</tr>
</tbody>
</table>

The undefined parameter shows up as NA in the coefficient table. The source of the singularity is revealed in the following table:

> with(soup, table(DAY, PRODID))

<table>
<thead>
<tr>
<th>PRODID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>369</td>
<td>94</td>
<td>184</td>
<td>93</td>
<td>88</td>
<td>93</td>
</tr>
<tr>
<td>2</td>
<td>370</td>
<td>275</td>
<td>0</td>
<td>92</td>
<td>97</td>
<td>92</td>
</tr>
</tbody>
</table>

which shows that the third PRODID was not presented at the second day at all. The design matrix will in this case be column rank deficient (also referred to as singular). This is detected by clm using the drop.coef function from ordinal. The following illustrates that the column rank of the design matrix is less than its number of columns:

> mm <- model.matrix(~ PRODID * DAY, data=soup)
> ncol(mm)
[1] 12
> qr(mm, LAPACK = FALSE)$rank
[1] 11

A similar type of rank deficiency occurs when variables in nominal are also present in formula or scale as illustrated in section 3.
9 Assessment of Model Convergence

The maximum likelihood estimates of the parameters in cumulative link models do not have closed form expressions, so iterative methods have to be applied to fit the models. Such iterative methods can fail to converge simply because an optimum cannot be found or because the parameter estimates are not determined accurately enough.

An optimum has been found if the maximum absolute gradient is small and if the condition number of the Hessian (defined here as the ratio of the largest to the smallest eigenvalues of the Hessian evaluated at convergence) is positive and not very large, say smaller than \(10^4\) or \(10^6\). The maximum absolute gradient (\texttt{max.grad}) and the condition number of the Hessian (\texttt{cond.H}) are reported by the summary method, for an example see page 4. A large condition number of the Hessian is not necessarily a problem, but it can be. A small condition number of the Hessian on the other hand is a good insurance that an optimum has been reached. Thus the maximum absolute gradient and the condition number of the Hessian can be used to check if the optimization has reached a well-defined optimum.

To determine the accuracy of the parameter estimates we use the \texttt{convergence} method:

```r
> convergence(fm1)
nobs logLik niter max.grad cond.H logLik.Error
72 -86.49 6(0) 4.01e-12 2.7e+01 <1e-10

Estimate Std.Err Gradient Error Cor.Dec Sig.Dig
1|2 -1.344 0.5171 2.06e-12 3.09e-13 12 13
2|3 1.251 0.4379 2.12e-12 -2.42e-13 12 13
3|4 3.467 0.5978 -4.01e-12 -9.32e-13 11 12
4|5 5.006 0.7309 -7.77e-14 -9.22e-13 11 12
tempwarm 2.503 0.5287 -4.53e-13 -6.33e-13 11 12
contactyes 1.528 0.4766 6.20e-14 -2.95e-13 12 13

Eigen values of Hessian:
21.7090 18.5615 10.3914 5.2093 4.0955 0.8163
```

The most important information is the number of correct decimals (\texttt{Cor.Dec}) and the number of significant digits (\texttt{Sig.Dig}) with which the parameters are determined. In this case all parameters are very accurately determined, so there is no reason to lower the convergence tolerance. The \texttt{logLik.error} shows that the error in the reported value of the log-likelihood is below \(10^{-10}\), which is by far small enough that likelihood ratio tests based on this model are accurate.

The convergence properties of the fitted model may be illustrated by plotting slices of the log-likelihood function for the parameters. The following code produce the slices in Figure 1.

```r
> slice.fm1 <- slice(fm1, lambda = 5)
> par(mfrow = c(2, 3))
> plot(slice.fm1)
```

The slices illustrates the log-likelihood function plotted as a function each parameter in turn while the remaining parameters are fixed at the ML estimates. The \texttt{lambda} argument controls how far from the ML estimates the slices should be computed; it can be interpreted as a multiplier in curvature units, where a curvature unit is similar to a standard error.

For an inspection of the log-likelihood function closer to the optimum we can use a smaller
Figure 1: Slices of the (negative) log-likelihood function for parameters in a model for the bitterness-of-wine data. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars indicate maximum likelihood estimates.
Figure 2: Slices of the log-likelihood function for parameters in a model for the bitterness-
of-wine data very close to the MLEs. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars the indicate maximum likelihood estimates.

lambda:
> slice2.fm1 <- slice(fm1, lambda = 1e-5)
> par(mfrow = c(2, 3))
> plot(slice2.fm1)

The resulting figure is shown in Fig. 2.

10 Profile Likelihood

The profile likelihood can be used for several things. Two of the most important objectives are to provide accurate likelihood confidence intervals and to illustrate effects of parameters in the fitted model.

Confidence intervals based on the profile likelihood were already obtained in section 2 and will not be treated any further here.

The effects of contact and temp can be illustrated with
> pr1 <- profile(fm1, alpha=1e-4)
> plot(pr1)

and provided in Figure 3. The alpha argument is the significance level controlling how far from the maximum likelihood estimate the likelihood function should be profiled. Learn more about the arguments to profile with help(profile.clm). From the relative profile
Figure 3: Relative profile likelihoods for the regression parameters in the Wine study. Horizontal lines indicate 95% and 99% confidence bounds.

likelihood for tempwarm we see that parameter values between 1 and 4 are reasonably well supported by the data, and values outside this range has little likelihood. Values between 2 and 3 are very well supported by the data and all have high likelihood.
References


A Tutorial on fitting Cumulative Link Mixed Models with \texttt{clmm2} from the \texttt{ordinal} Package.

Christensen, R. H. B. (2012) A Tutorial on fitting Cumulative Link Mixed Models with \texttt{clmm2} from the \texttt{ordinal} Package.
A Tutorial on fitting Cumulative Link Mixed Models with \texttt{clmm2} from the \texttt{ordinal} Package

Rune Haubo B Christensen

September 10, 2012

Abstract

It is shown by example how a cumulative link mixed model is fitted with the \texttt{clmm2} function in package \texttt{ordinal}. Model interpretation and inference is briefly discussed. A tutorial for the more recent \texttt{clmm} function is work in progress.

We will consider the data on the bitterness of wine from Randall (1989) presented in Table 1 and available as the object \texttt{wine} in package \texttt{ordinal}. The data were also analyzed with mixed effects models by Tutz and Hennevogl (1996). The following gives an impression of the wine data object:

\begin{verbatim}
> data(wine)
> head(wine)

response rating temp contact bottle judge
1 36 2 cold no 1 1
2 48 3 cold no 2 1
3 47 3 cold yes 3 1
4 67 4 cold yes 4 1
5 77 4 warm no 5 1
6 60 4 warm no 6 1

> str(wine)

'data.frame': 72 obs. of 6 variables:
$ response: num 36 48 47 67 77 60 83 90 17 22 ... 
$ rating : Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<..: 2 3 3 4 4 4 5 5 1 2 ... 
$ temp : Factor w/ 2 levels "cold","warm": 1 1 1 1 2 2 2 2 1 1 ... 
$ contact : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 2 2 1 1 ... 
$ bottle : Factor w/ 8 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 1 2 ... 
$ judge : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 2 2 ... 

The data represent a factorial experiment on factors determining the bitterness of wine with 1 = “least bitter” and 5 = “most bitter”. Two treatment factors (temperature and contact) each have two levels. Temperature and contact between juice and skins can be controlled when crushing grapes during wine production. Nine judges each assessed wine from two bottles from each of the four treatment conditions, hence there are 72 observations in all. For more information see the manual entry for the wine data: \texttt{help(wine)}.

We will fit the following cumulative link mixed model to the wine data:

\[
\text{logit}(P(Y_i \leq j)) = \theta_j - \beta_1(\text{temp}_i) - \beta_2(\text{contact}_i) - u(\text{judge}_i)
\]

\begin{equation}
\text{logit}(P(Y_i \leq j)) = \theta_j - \beta_1(\text{temp}_i) - \beta_2(\text{contact}_i) - u(\text{judge}_i)
\end{equation}

(1)
This is a model for the cumulative probability of the $i$th rating falling in the $j$th category or below, where $i$ index all observations and $j = 1, \ldots, J$ index the response categories $(J = 5)$. \{\theta_j\}$ are known as threshold parameters or cut-points. We take the judge effects to be random, and assume that the judge effects are IID normal: $u(\text{judge}_i) \sim N(0, \sigma^2_u)$.

We fit this model with the \texttt{clmm2} function in package \texttt{ordinal}. Here we save the fitted \texttt{clmm2} model in the object \texttt{fm1} (short for \texttt{fitted model 1}) and print the model by simply typing its name:

```r
> fm1 <- clmm2(rating ~ temp + contact, random=judge, data=wine)
> fm1
```

Cumulative Link Mixed Model fitted with the Laplace approximation

Call:
clmm2(location = rating ~ temp + contact, random = judge, data = wine)

Random effects:
Var Std.Dev
judge 1.279455 1.13113

Location coefficients:
tempwarm contactyes
3.062993 1.834883

No Scale coefficients

Threshold coefficients:
  1|2 2|3 3|4 4|5
-1.623664 1.513364 4.228525 6.088770

log-likelihood: -81.56541
AIC: 177.1308

Maximum likelihood estimates of the parameters are provided using the Laplace approximation to compute the likelihood function. A more accurate approximation is provided by the adaptive Gauss-Hermite quadrature method. Here we use 10 quadrature nodes and use the \texttt{summary} method to display additional information:

```r
> fm2 <- clmm2(rating ~ temp + contact, random=judge, data=wine,

This table shows the ratings of the bitterness of some white wines. Data are adopted from Randall (1989).

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Contact</th>
<th>Bottle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>cold</td>
<td>no</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>cold</td>
<td>no</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>cold</td>
<td>yes</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>cold</td>
<td>yes</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>warm</td>
<td>no</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>warm</td>
<td>no</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>warm</td>
<td>yes</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>warm</td>
<td>yes</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
> summary(fm2)
Cumulative Link Mixed Model fitted with the adaptive Gauss-Hermite
quadrature approximation with 10 quadrature points

Call:
clmm2(location = rating ~ temp + contact, random = judge, data = wine,
Hess = TRUE, nAGQ = 10)

Random effects:
  Var Std.Dev
judge 1.287741 1.134787

Location coefficients:
  Estimate Std. Error z value Pr(>|z|)
tempwarm 3.0619 0.5951 5.1454 2.6700e-07
tcontactyes 1.8334 0.5122 3.5797 0.00034398

No scale coefficients

Threshold coefficients:
  Estimate Std. Error z value
1|2 -1.6235 0.6834 -2.3757
2|3 1.5128 0.6044 2.5028
3|4 4.2271 0.8090 5.2251
4|5 6.0862 0.9719 6.2618

log-likelihood: -81.53246
AIC: 177.0649
Condition number of Hessian: 27.62149

The small changes in the parameter estimates show that the Laplace approximation was
in fact rather accurate in this case. Observe that we set the option Hess = TRUE. This
is needed if we want to use the summary method since the Hessian is needed to compute
standard errors of the model coefficients.

The results contain the maximum likelihood estimates of the parameters:

$$\hat{\beta}_1 = 3.06, \quad \hat{\beta}_2 = 1.83, \quad \hat{\sigma}_u^2 = 1.29 = 1.13^2, \quad \{\hat{\theta}_j\} = [-1.62, 1.51, 4.23, 6.09]. \quad (2)$$

Observe the number under Std.Dev for the random effect is not the standard error of the
random effects variance, Var. Rather, it is the standard deviation of the random effects, i.e.,
it is the square root of the variance. In our example $\sqrt{1.29} \approx 1.13$.

The condition number of the Hessian measures the empirical identifiability of the model.
High numbers, say larger than $10^4$ or $10^6$ indicate that the model is ill defined. This would
indicate that the model can be simplified, that possibly some parameters are not identifiable,
and that optimization of the model can be difficult. In this case the condition number of
the Hessian does not indicate a problem with the model.

The coefficients for temp and contact are positive indicating that higher temperature and
more contact increase the bitterness of wine, i.e., rating in higher categories is more likely.
The odds ratio of the event $Y \geq j$ is $\exp(\hat{\beta}_{\text{treatment}})$, thus the odds ratio of bitterness being
rated in category $j$ or above at warm relative to cold temperatures is
> \exp(\text{coef(fm2)[5]})

tempwarm
21.36805

The p-values for the location coefficients provided by the \texttt{summary} method are based on the so-called Wald statistic. More accurate test are provided by likelihood ratio tests. These can be obtained with the \texttt{anova} method, for example, the likelihood ratio test of \texttt{contact}

\begin{verbatim}
> fm3 <- clmm2(rating ~ temp, random=judge, data=wine, nAGQ=10)
> anova(fm3, fm2)
\end{verbatim}

Likelihood ratio tests of cumulative link models

| Response: rating |
|------------------|------------------|------------------|------------------|
| Model | Resid. df | -2logLik | Test Df LR stat. | Pr(Chi) |
| temp | 66 | 177.4090 | | |
| temp + contact | 65 | 163.0649 | 1 vs 2 14.34409 0.0001522572 |

which in this case is slightly more significant. The Wald test is not reliable for variance parameters, so the \texttt{summary} method does not provide a test of $\sigma_u$, but a likelihood ratio test can be obtained with \texttt{anova}:

\begin{verbatim}
> fm4 <- clm2(rating ~ temp + contact, data=wine)
> anova(fm4, fm2)
\end{verbatim}

Likelihood ratio tests of cumulative link models

| Response: rating |
|------------------|------------------|------------------|------------------|
| Model | Resid. df | -2logLik | Test Df LR stat. | Pr(Chi) |
| temp + contact | 66 | 172.9838 | | |
| temp + contact | 65 | 163.0649 | 1 vs 2 9.918925 0.001635879 |

showing that the judge term is significant. Since this test of $\sigma_u = 0$ is on the boundary of the parameter space (a variance cannot be negative), it is often argued that a more correct p-value is obtained by halving the p-value produced by the conventional likelihood ratio test. In this case halving the p-value is of little relevance.

A profile likelihood confidence interval of $\sigma_u$ is obtained with:

\begin{verbatim}
> pr2 <- profile(fm2, range=c(.1, 4), nSteps=30, trace=0)
> confint(pr2)
\end{verbatim}

<table>
<thead>
<tr>
<th>2.5 %</th>
<th>97.5 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>stDev 0.5048597 2.272327</td>
<td></td>
</tr>
</tbody>
</table>

The profile likelihood can also be plotted:

\begin{verbatim}
> plot(pr2)
\end{verbatim}

The result is shown in Fig. 1 where horizontal lines indicate 95% and 99% confidence bounds. Clearly the profile likelihood function is asymmetric and symmetric confidence intervals would be inaccurate.

The judge effects, $u(judge_i)$ are not parameters, so they cannot be estimated in the conventional sense, but a “best guess” is provided by the conditional modes. Similarly the \textit{conditional variance} provides an uncertainty measure of the conditional modes. These quantities are included in \texttt{clmm2} objects as the \texttt{ranef} and \texttt{condVar} components. The following code generates the plot in Fig. 2 illustrating judge effects via conditional modes with 95%
confidence intervals based on the conditional variance:

```r
ci <- fm2$ranef + qnorm(0.975) * sqrt(fm2$condVar) %o% c(-1, 1)
ord.re <- order(fm2$ranef)
ci <- ci[order(fm2$ranef),]
plot(1:9, fm2$ranef[ord.re], axes=FALSE, ylim=range(ci),
     xlab="Judge", ylab="Judge effect")
axis(1, at=1:9, labels = ord.re)
axis(2)
for(i in 1:9) segments(i, ci[i,1], i, ci[i, 2])
abline(h = 0, lty=2)
```

The seventh judge gave the lowest ratings of bitterness while the first judge gave the highest ratings of bitterness. The significant judge effect indicate that judges perceived the bitterness of the wines differently. Two natural interpretations are that either a bitterness of, say, 3 means different things to different judges, or the judges actually perceived the bitterness of the wines differently. Possibly both effects play their part.

The fitted or predicted probabilities can be obtained with the judge effects at their conditional modes or for an average judge (μ = 0). The former are available with `fitted(fm)` or with `predict(fm)`, where `fm` is a fitted model object. In our example we get

```r
> head(cbind(wine, fitted(fm2)))
response rating temp contact bottle judge fitted(fm2)
1 36 2 cold no 1 1 0.4188842
2 48 3 cold no 2 1 0.4724113
3 47 3 cold yes 3 1 0.5499094
4 67 4 cold yes 4 1 0.2607354
5 77 4 warm no 5 1 0.4203997
6 60 4 warm no 6 1 0.4203997
```

Predicted probabilities for an average judge can be obtained by including the data used to fit the model in the `newdata` argument of `predict:`
Figure 2: Judge effects given by conditional modes with 95% confidence intervals based on the conditional variance.

> head(cbind(wine, pred=predict(fm2, newdata=wine)))

<table>
<thead>
<tr>
<th>response</th>
<th>rating</th>
<th>temp</th>
<th>contact</th>
<th>bottle</th>
<th>judge</th>
<th>pred</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36</td>
<td>2</td>
<td>no</td>
<td>1</td>
<td>1</td>
<td>0.6547512</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td>3</td>
<td>no</td>
<td>2</td>
<td>1</td>
<td>0.1661397</td>
</tr>
<tr>
<td>3</td>
<td>47</td>
<td>3</td>
<td>yes</td>
<td>3</td>
<td>1</td>
<td>0.4958192</td>
</tr>
<tr>
<td>4</td>
<td>67</td>
<td>4</td>
<td>yes</td>
<td>4</td>
<td>1</td>
<td>0.0696343</td>
</tr>
<tr>
<td>5</td>
<td>77</td>
<td>4</td>
<td>warm</td>
<td>no</td>
<td>5</td>
<td>1.0913894</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
<td>4</td>
<td>warm</td>
<td>no</td>
<td>6</td>
<td>1.0913894</td>
</tr>
</tbody>
</table>

Model (1) says that for an average judge at cold temperature the cumulative probability of a bitterness rating in category $j$ or below is

$$P(Y_i \leq j) = \logit^{-1}[\theta_j - \beta_2(\text{contact}_i)]$$

since $u$ is set to zero and $\beta_1(\text{temp}_i) = 0$ at cold conditions. Further, $\logit^{-1}(\eta) = 1/[1 + \exp(\eta)]$ is the cumulative distribution function of the logistic distribution available as the \texttt{plogis} function. The (non-cumulative) probability of a bitterness rating in category $j$ is

$$\pi_j = P(Y_i \leq j) - P(Y_i \leq j - 1),$$

for instance the probability of a bitterness rating in the third category at these conditions can be computed as

> plogis(fm2$Theta[3] - fm2$beta[2]) -
plogis(fm2$Theta[2] - fm2$beta[2])

3|4
0.4958192

This corresponds to the third entry of \texttt{predict(fm2, newdata=wine)} given above.

Judge effects are random and normally distributed, so an average judge effect is 0. Extreme judge effects, say 5th and 95th percentile judge effects are given by

> qnorm(0.95) * c(-1, 1) * fm2$stDev

[1] -1.866558 1.866558
At the baseline experimental conditions (cold and no contact) the probabilities of bitterness ratings in the five categories for a 5th percentile judge is

```r
> pred <-
  function(eta, theta, cat = 1:(length(theta)+1), inv.link = plogis)
  {
    Theta <- c(-1e3, theta, 1e3)
    sapply(cat, function(j)
      inv.link(Theta[j+1] - eta) - inv.link(Theta[j] - eta) )
  }
> pred(qnorm(0.05) * fm2$stDev, fm2$Theta)

1|2  2|3  3|4  4|5
0.5604689932 0.4065840002 0.0306948522 0.0019005715 0.0003515829
```

We can compute these probabilities for average, 5th and 95th percentile judges at the four experimental conditions. The following code plots these probabilities and the results are shown in Fig. 3.

```r
> mat <- expand.grid(judge = qnorm(0.95) * c(-1, 0, 1) * fm2$stDev,
                         contact = c(0, fm2$beta[2]),
                         temp = c(0, fm2$beta[1]))
> pred.mat <- pred(eta=rowSums(mat), theta=fm2$Theta)
> lab <- paste("contact=" , rep(levels(wine$contact), 2), ", ",
                   "temp=" , rep(levels(wine$temp), each=2), sep=""
> par(mfrow=c(2, 2))
> for(k in c(1, 4, 7, 10)) {
    plot(1:5, pred.mat[k,], lty=2, type = "l", ylim=c(0,1),
         xlab="Bitterness rating scale", axes=FALSE,
         ylab="Probability", main=lab[ceiling(k/3)], las=1)
    axis(1); axis(2)
    lines(1:5, pred.mat[k+1, ], lty=1)
    lines(1:5, pred.mat[k+2, ], lty=3)
    legend("topright",
            c("avg. judge", "5th %-tile judge", "95th %-tile judge"),
            lty=1:3, bty="n")
}
```

At constant experimental conditions the odds ratio for a bitterness rating in category \( j \) or above for a 95th percentile judge relative to a 5th percentile judge is

```r
> exp(2*qnorm(0.95) * fm2$stDev)

judge
41.80921
```

The differences between judges can also be expressed in terms of the interquartile range: the odds ratio for a bitterness rating in category \( j \) or above for a third quartile judge relative to a first quartile judge is

```r
> exp(2*qnorm(0.75) * fm2$stDev)

judge
4.621893
```
Figure 3: Rating probabilities for average and extreme judges at different experimental conditions.
References


Appendix M

Statistical methodology for sensory discrimination tests and its implementation in sensR

Statistical methodology for sensory discrimination tests
and its implementation in sensR

Rune Haubo Bojesen Christensen

April 23, 2012

Abstract

The statistical methodology of sensory discrimination analysis is described. This forms the basis of the implementation in the sensR package for R. Implementation choices will be motivated when appropriate and examples of analysis of sensory discrimination experiments will be given throughout using the sensR package. This document currently covers parameterizations, hypothesis tests, confidence intervals, and power and sample size calculations for the four common discrimination protocols: 2-AFC, 3-AFC, triangle and duo-trio; analysis of replicated experiments with the four common discrimination protocols using the beta-binomial and chance-corrected beta-binomial models.
Contents

1 Introduction ............................................. 3

2 Classification of sensory discrimination protocols ......... 3

3 Four common sensory discrimination protocols: 2-AFC, 3-AFC, triangle and duo-trio ......... 4
   3.0.1 Implementation in sensR ......................... 7

3.1 Inference in simple discrimination protocols ............... 7
   3.1.1 Standard errors .................................. 8
   3.1.2 The likelihood function ............................ 8
   3.1.3 Confidence intervals ............................... 9
   3.1.4 Sensory difference tests ........................... 10
   3.1.5 Sensory similarity tests ............................ 11
   3.1.6 Confidence intervals and hypothesis tests .......... 11
   3.1.7 Implementation in sensR ......................... 12

3.2 Sample size and power calculations for simple discrimination protocols ... 15
   3.2.1 The critical value .................................. 15
   3.2.2 The power of difference tests ..................... 16
   3.2.3 The power of similarity tests ..................... 17
   3.2.4 Power calculation based on simulations ............ 18
   3.2.5 Power calculation based on the normal approximation .... 19
   3.2.6 Sample size determination ........................ 20
1 Introduction

The aim of this document is 1) to describe the statistical methodology for sensory discrimination testing and analysis, and 2) to describe how such analyses can be performed in R using package sensR (Christensen and Brockhoff, 2010) co-developed by the author of this document.

This document is divided into sections that cover topics with similar statistical methodology. Implementation choices in the sensR package will be described in connection with the statistical methodology whenever appropriate. Small examples illustrating the use of functions in the sensR package will appear throughout.

This is not a hands-on practical tutorial to analysis of sensory discrimination experiments with the sensR package, neither is it a user friendly introduction to discrimination and similarity testing in sensory discrimination protocols. The former document does not really exist1 (yet), and for the latter document, we refer the reader to (Næs et al., 2010, chapter 7). We will assume throughout that the reader has basic statistical training and is familiar with sensory discrimination testing to the level of (Næs et al., 2010, chapter 7).

2 Classification of sensory discrimination protocols

The most common and simplest discrimination protocols comprise the 2-AFC, 3-AFC, triangle, duo-trio, A-not A and same-different protocols. The first four protocols are designed such that the response follows a binomial distribution in the simplest experimental setting. On the other hand responses from A-not A and same-different protocols are distributed according to a compound or product binomial distribution in the simplest experimental setting. An extension of the A-not A method known as the A-not A with sureness is a classical SDT method which leads to multinomially distributed responses. Similarly the same-different method extends to the degree-of-difference protocol also resulting in multinomially distributed responses. An experiment using one of the first four simple protocols can be summarized with the proportion of correct responses or similarly the probability of discrimination or d-prime. The Thurstonian models for the remaining protocols involve one or more additional parameters each with their particular cognitive interpretation.

The 2-AFC and 3-AFC protocols are so-called directional protocols since they require that the nature of the difference (e.g. sweetness) is provided as part of the assessor instructions. On the other hand the triangle and duo-trio protocols are not directional since these protocols are used to test un-specified differences. From a Thurstonian point of view, the sensory dimension or the perceptual dimension is fixed in the 2-AFC and 3-AFC methods. The cognitive decision strategy is consequently assumed different in these two classes of protocols. When the perceptual dimension is fixed, the assessors may use the more effective skimming strategy, while assessors are forced to use the inferior comparison of distances strategy when using the un-directional protocols.

The A-not A and same-different protocols are methods with so-called response bias. Response bias refers to the concept that one type of response is preferred over another despite the sensory distance remains unchanged. For instance some assessors may prefer the “A” response over the “not A” response.

---

1this is on the to-do list of the author of this document, so there is hope it will appear in the future.
The four simple protocols are without response bias since no response can be consistently preferred over another without affecting the discriminative effect. The decision criterion is said to be fixed or stabilized.

3 Four common sensory discrimination protocols: 2-AFC, 3-AFC, triangle and duo-trio

The four common sensory discrimination protocols are often used in practical applications in the food industry as well as in other areas. They are also of considerable interest in the scientific literature about sensory discrimination.

The protocols have one important thing in common from a statistical perspective: their statistical models can all be described as variants of the binomial distribution. That is, the answer from any one of these protocols is either correct or incorrect and the sampling distribution of answers is therefore a binomial distribution.

For the duo-trio and 2-AFC protocols the guessing probability, \( p_g \) is 1/2. This means that if there is no discriminative difference between the products, then the probability of a correct answers, \( p_c \) is one half. Similarly for the triangle and 3-AFC protocols the guessing probability is 1/3. The four common discrimination protocols are said to be free of response bias in contrast to the A-not A and same-different protocols.

If we assume for a moment that the population of assessors (be that judges in an expert panel or consumers) is comprised of ignorants who are always guessing and discriminators who always discriminate correctly and provide the appropriate answer (though this will not always be the correct answer). One way to express the sensory distance of the objects (or discriminative ability of the assessors — we will treat these viewpoints synonymously throughout) is the proportion of discriminators, \( p_d \) in the population of interest. It is almost always an unreasonable assumption that some assessors are either always discriminating or always guessing (Ennis, 1993), but we may still talk about the probability of discrimination. This probability may refer to particular individuals or to a population; in this section we will adopt a population perspective.

The relation between the probability of a correct answer and the probability of discrimination is

\[
 p_c = p_g + p_d (1 - p_g),
\]

(1)

where the guessing probability, \( p_g \) is 1/2 for the duo-trio and 2-AFC protocols and 1/3 for the triangle and 3-AFC protocols. The reverse relation is

\[
 p_d = (p_c - p_g) / (1 - p_g).
\]

(2)

Another way to summarize the sensory distance is through a measure known as \( d' \) (pronounced “d-prime”) from signal detection theory (SDT, Green and Swets, 1966; Macmillan and Creelman, 2005), or equivalently the Thurstonian delta, \( \delta \) (Thurstone, 1927a,b,c). These two concepts are identical and will be used synonymously throughout, and they are actually based on the same underlying psychophysical model for the cognitive process. Whereas \( p_c \) is a measure and parameter completely free of reference to any particular discrimination protocol, \( p_d \) depends on the discrimination protocol through the guessing probability, but \( d' \) depends on the discrimination protocol through the so-called psychometric function, for
the discrimination protocol. The psychometric function maps from $d'$ to the probability of a correct answer:

$$p_c = f_{ps}(d').$$

For the $m$-AFC method, where $m$ denotes the number of “forced choices”, the psychometric function is given by

$$f_{m\text{-AFC}}(d') = \int_{-\infty}^{\infty} \phi(z - d')\Phi(z)^{m-1} \, dz,$$

where $\phi$ is the standard normal probability density function and $\Phi$ is the standard normal cumulative distribution function. The psychometric functions for the four common discrimination protocols are given by

$$f_{3\text{-AFC}}(d') = \int_{-\infty}^{\infty} \phi(z - d')\Phi(z)^2 \, dz$$

$$f_{2\text{-AFC}}(d') = \int_{-\infty}^{\infty} \phi(z - d')\Phi(z) \, dz = \Phi(d'/\sqrt{2})$$

$$f_{\text{tri}}(d') = 2 \int_0^{\infty} \left\{ \Phi\left[-z/\sqrt{3} + d'/\sqrt{2}/3\right] + \Phi\left[-z/\sqrt{3} - d'/\sqrt{2}/3\right] \right\} \phi(z) \, dz$$

$$f_{\text{duo-trio}}(d') = 1 - \Phi(d'/\sqrt{2}) - \Phi(d'/\sqrt{6}) + 2\Phi(d'/\sqrt{2})\Phi(d'/\sqrt{6}).$$

Further information can be found in Ennis (1993) and Brockhoff and Christensen (2010).

The relations between the three scales at which a sensory difference is described are illustrated in Fig. 1. In the relation between $p_d$ and $d'$ the alternative forced choice protocols behave similarly, while the duo-trio and triangle protocols behave similarly. The gradient of the psychometric functions (cf. eq. (17)) goes to zero when $d'$ goes to zero for the duo-trio and triangle protocols.

The result of a simple discrimination protocol is a number of correct answers, $X = x$ out of $n$ trials. Under the assumption of independent observations, the sampling distribution of $X$ is the binomial:

$$X \sim \text{Binom}(p_c, n),$$

so

$$P(X = x) = \binom{n}{x} p_c^x (1 - p_c)^{n-x}. \quad (10)$$

There is a subtle but important distinction between the proportion of a correct answer and the probability of a correct answer. The proportion of correct answers is $x/n$ which can be any number between 0 and 1. The probability of a correct answer, which we denote by $p_c$, is a parameter and represents a true underlying value. As such $p_c$ cannot be lower than the guessing probability for the discrimination protocol that was used and cannot exceed 1. The usual estimator of a binomial probability is just the sample proportion, $x/n$, but this is not the case here, and it is exactly this feature that makes discrimination testing interesting statistically.

The maximum likelihood (ML) estimator\(^2\) of $p_c$ is given by:

$$\hat{p}_c = \begin{cases} 
\frac{x}{n} & \text{if } \frac{x}{n} \geq p_g \\
 p_g & \text{if } \frac{x}{n} < p_g 
\end{cases} \quad (11)$$

\(^2\)Following standard statistical practice we use the hat-notation to denote an estimator or an estimate
Figure 1: The connection between $d'$, $P_c$, and $P_d$ for the four common sensory discrimination protocols. The so-called psychometric functions; $P_c$ as a function of $d'$, are shown in the upper left figure.
The ML estimator of $p_d$ is given by application of eq. (2), and the ML estimator of $d'$, by inversion of eq. (3), given by

$$
\hat{d} = f_{ps}^{-1}(\hat{p}_c),
$$

(12)

where $f_{ps}^{-1}(\cdot)$ (which should not be confused with $f_{ps}(\cdot)^{-1} = 1/f_{ps}(\cdot)$) is the inverse psychometric function.

The allowed ranges (parameter space) for these three parameters are given by

$$
d' \in [0, \infty[, \quad p_d \in [0, 1], \quad p_c \in [p_g, 1].
$$

(13)

Negative $d'$ values are sometimes mentioned in the literature, but negative $d'$ values are not possible in the discrimination protocols that we consider here. They are possible in preference tests and theoretically possible in Thurstonian models based on other assumptions, see section XXX for more background information on this topic.

### 3.0.1 Implementation in sensR

In package sensR there is a function `rescale` that maps between the three scales; $p_c$, $p_d$ and $d'$. A value on one of these scales is given as argument and values on all three scales are given in the results. The results respect the allowed ranges of the parameters in eq. (13), so if the supplied $p_c$ is less than $p_g$, then $p_c = p_g$ is returned with $p_d$ and $d'$ at the appropriate levels:

```r
> rescale(pc = 0.25, method = "triangle")
```

Estimates for the triangle protocol:

<table>
<thead>
<tr>
<th>pc</th>
<th>pd</th>
<th>d.prime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3333333</td>
<td>0</td>
</tr>
</tbody>
</table>

Function `rescale` use a number of auxiliary functions for its computations; these are also directly available to the package user:

- `pc2pd`: maps from the $p_c$-scale to the $p_d$-scale.
- `pd2pc`: maps from the $p_d$-scale to the $p_c$-scale.
- `psyfun`: implements the psychometric functions $p_c = f_{ps}(d')$ for the four common discrimination protocols, cf. eq. (3).
- `psyinv`: implements the inverse psychometric functions, $d' = f_{ps}^{-1}(p_c)$ for the four common discrimination protocols, cf. eq. (12).
- `psyderiv`: implements the derivative of the psychometric functions, $f_{ps}'(d')$ for the four common discrimination protocols.

### 3.1 Inference in simple discrimination protocols

To obtain inference in simple discrimination protocols, we need measures such as standard errors, confidence intervals (CIs) and $p$-values from significance tests.
3.1.1 Standard errors

The standard error of \( p_c \) is given by:

\[
se(p_c) = \sqrt{p_c(1 - p_c)/n}. \tag{14}
\]

The standard error of \( p_d \) and \( d' \) can be found by application of the Delta method (see for example Pawitan, 2001):

\[
se\{f(x)\} = \frac{\partial f(x)}{\partial x} se(x) \tag{15}
\]

The standard error of \( p_d \) is therefore

\[
se(p_d) = \frac{1}{1 - p_g} se(p_c) \tag{16}
\]

since \( \partial p_d / \partial p_c = 1/(1 - p_g) \), cf. eq. (2). The standard error of \( d' \) can similarly be found as

\[
se(d') = \frac{\partial f_{ps}^{-1}(p_c)}{\partial p_c} se(p_c) = \frac{1}{f_{ps}'(d')} se(p_c) \tag{17}
\]

where \( f_{ps}'(d') \) is the derivative of the psychometric function with respect to \( d' \); expressions are given by Brockhoff and Christensen (2010).

Standard errors are only defined and only meaningful as measures of uncertainty when the parameter estimate is at the interior of the parameter space, i.e. when the parameter estimate is not at the boundary of its allowed range, cf. eq. (13).

Even when the parameter estimate is close, in some sense, to the boundary of its parameter space, the standard error is not a meaningful measure of uncertainty, because the uncertainty is in fact asymmetric. This means that symmetric confidence intervals based on the standard error will also be misleading and other techniques should be applied.

3.1.2 The likelihood function

The (log-)likelihood function can be used to obtain likelihood ratio or likelihood root statistics for hypothesis tests, and it can be used to construct confidence intervals with good properties.

The log-likelihood function for a model based on the binomial distribution is given by

\[
\ell(p_c; x, n) = C + x \log p_c + (n - x) \log(1 - p_c), \tag{18}
\]

where \( C = \log \binom{n}{x} \) is a constant with respect to \( p_c \). The log-likelihood function for \( p_d \) or \( d' \) is given by combining eq. (18) with (2) or (12).

In general, standard errors can be found as the square root of the diagonal elements of the variance-covariance matrix of the parameters. The variance-covariance matrix can be found as the inverse of the negative Hessian matrix (the matrix of second order derivatives) of the log-likelihood function evaluated at the ML estimates. Here there is only one parameter (either one of \( p_c \), \( p_d \) or \( d' \)), so the matrices are merely scalars.

It can be shown that the same standard errors as those derived in eq. (14), (16) and (17) can be derived by differentiating (18) twice and using the chain rule to obtain the standard errors of \( p_d \) and \( d' \).
3.1.3 Confidence intervals

There are several general approaches to get CIs for parameters. One general way that applies (with varying success) to almost all parameters with a standard error is the traditional Wald interval:

\[ CI : \hat{\mu} \pm z_{1-\alpha/2}se(\hat{\mu}), \]  

(19)

where \( z_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2) \) is the upper \( \alpha/2 \) quantile of the standard normal distribution. This CI is based on the Wald statistic\(^3\):

\[ w(\mu_0) = (\hat{\mu} - \mu_0)/se(\hat{\mu}). \]  

(20)

The CI may also be expressed more generally for a statistic \( t(\mu_0) \) that follows a standard normal distribution under the null hypothesis as:

\[ CI : \{ \mu ; \mid t(\mu) \mid < z_{1-\alpha/2} \}. \]  

(21)

Using \( w \) as \( t \) in (21) gives the interval (19).

Another general approach is to use the likelihood root statistic (inverted likelihood ratio test) which applies to all likelihood based models and almost always impressively successful. The likelihood root statistic is given by:

\[ r(\mu_0) = \text{sign}(\hat{\mu} - \mu_0) \sqrt{2 \{ \ell(\hat{\mu};x) - \ell(\mu_0;x) \}} \]  

(22)

Both the Wald and likelihood root statistics asymptotically follow standard normal distributions under the null hypothesis. Even though their asymptotic behavior is in fact identical, their finite sample properties may be quite different and often favor the likelihood root statistic since it removes nonlinear parameterization effects.

A disadvantage of Wald intervals is that they are not invariant to nonlinear transformations of the parameter. This means that a Wald CI for \( p_c \) and a Wald CI for \( d' \) provides different kinds of evidence about the parameters and could, for instance, lead to inclusion of \( p_y \) in the CI on the \( p_c \) scale, but exclusion of \( d' = 0 \) on the \( d' \) scale. More generally the Wald CI for \( p_c \) cannot be found by transforming the Wald CI limits for \( d' \) through the psychometric function. The CIs based on the likelihood root statistic is on the other hand invariant to nonlinear transformations of the parameter. This means the likelihood CI for \( d' \) can be found by either computing the likelihood CI for \( d' \) directly or by transforming the limits of the likelihood CI for \( p_c \) through the inverse psychometric function — they give the same answer. The evidence provided by the likelihood CI is therefore invariant to the choice of scale.

Another approach to generate CIs consistent across parameter scales would be to compute an appropriate CI for, say, \( p_c \) and then transform the CI limits through the appropriate functions to obtain CIs for \( p_y \) and \( d' \). For likelihood CIs this does not make any difference, of course. If an appropriate CI can be computed on any one scale, this would provide appropriate CIs on the other scales as well. There exists a wide range of CIs for the binomial probability parameter (refs), for instance the score interval and the so-called exact interval in addition to the Wald and likelihood intervals.

The ‘exact’ binomial interval is given by inversion of the ‘exact’ binomial test and known as the Clopper-Pearson interval (Clopper and Pearson, 1934). The lower and upper limits are

\(^3\)actually the original definition used \( se(\mu_0) \) in the denominator.
defined as the values of $p_c$ that solve:

$$LL : P(X \geq x) = \alpha/2, \quad UL : P(X \leq x) = \alpha/2,$$

(23)

where $X \sim \text{binom}(p_c, n)$. Rather than solving these equations numerically, the limits can be found directly as quantiles of the beta distribution, $\text{Beta}(a, b)$: the lower limit is the $\alpha/2$ quantile of $\text{Beta}(x, n-x+1)$ and the upper limit is the $1-\alpha/2$ quantile of $\text{Beta}(x+1, n-x)$.

Another commonly applied statistic is based on the normal approximation of the binomial distribution. Asymptotically $(X - np_c)/\sqrt{np_c(1-p_c)}$ behaves like a standard normal random variable, so we may use

$$w^*(p_c0) = \frac{x - np_c0}{\sqrt{np_c0(1-p_c0)}},$$

(24)
as test statistic. This statistic is in fact identical to the Wald statistic (20) if $\text{se}(\mu_0)$ is used in the denominator instead of $\text{se}(\hat{\mu})$.

The statistic $w^*$ is related to the Pearson $\chi^2$ statistic

$$X^2(p_c0) = \frac{(x - np_c0)^2}{np_c0} + \frac{(n - x - n(1-p_c0))^2}{n(1-p_c0)}$$

(25)
since $w^*$ is the signed square root of $X^2$. Similarly the likelihood root statistic, $r(p_c0)$ is related to the likelihood ratio statistic

$$G^2(p_c0) = x \log \frac{x}{np_c0} + (n - x) \log \frac{n - x}{n(1-p_c0)}$$

(26)
since $r(p_c0)$ is the signed square root of $G^2(p_c0)$.

### 3.1.4 Sensory difference tests

A sensory difference test is a test of

$$H_0 : \begin{cases} p_c \leq p_{c0} \\
\quad p_d \leq p_{d0} \\
d' \leq d'_{00}
\end{cases} \quad \text{versus} \quad H_A : \begin{cases} p_c > p_{c0} \\
\quad p_d > p_{d0} \\
d' > d'_{00}
\end{cases},$$

(27)

where the traditional tests of no-difference is given by choosing $p_{c0} = p_g$, $p_{d0} = 0$ and $d'_{00} = 0$ making the null hypothesis an equality rather than an inequality.

The $p$-value of a difference test is the probability of observing a number of successes that is as large or larger than that observed given the null hypothesis that the probability of a correct answer is $p_{c0}$. The $p$-value based on the ‘exact’ binomial test is therefore:

$$p\text{-value} = P(X \geq x) = 1 - \sum_{i=0}^{x-1} \binom{n}{i} p_{c0}^i (1-p_{c0})^{n-i},$$

(28)

where $X \sim \text{binom}(p_{c0}, n)$.

The $p$-value for a difference based on a statistic, $t(\mu_0)$ that follows a standard normal distribution under the null hypothesis is given by:

$$p\text{-value} = P\{Z \geq t(\mu_0)\} = 1 - \Phi\{t(\mu_0)\},$$

(29)

where $Z$ is a standard normal random variable and $\Phi$ is the standard normal cumulative distribution function.
3.1.5 Sensory similarity tests

A sensory similarity test is a test of

\[ H_0 : \begin{align*}
  p_c & \geq p_{c0} \\
p_d & \geq p_{d0} \\
d' & \geq d'_0
\end{align*} \quad \text{versus} \quad H_A : \begin{align*}
  p_c & < p_{c0} \\
p_d & < p_{d0} \\
d' & < d'_0
\end{align*}, \tag{30}
\]

where subject matter considerations and possibly power computations will guide the choice of \( p_{c0}, p_{d0} \) or \( d'_0 \). Observe that \( d'_0 \) has to be positive for the test to make sense.

The \( p \)-value of a similarity test is the probability of observing a number of successes that is as large or less than that observed given the null hypothesis that the probability of a correct answer is \( p_{c0} \). The \( p \)-value based on the ‘exact’ binomial test is therefore:

\[ p\text{-value} = P(X \leq x) = \sum_{i=0}^{x} \binom{n}{i} p_{c0}^i (1 - p_{c0})^{n-i}, \tag{31} \]

where \( X \sim \text{binom}(p_{c0}, n) \)

The \( p \)-value for a difference based on a statistic, \( t(\mu_0) \) that follows a standard normal distribution under the null hypothesis is given by:

\[ p\text{-value} = P\{Z \leq t(\mu_0)\} = \Phi\{t(\mu_0)\}, \tag{32} \]

3.1.6 Confidence intervals and hypothesis tests

Confidence intervals are often described by their relation to hypothesis tests such that a two-sided hypothesis test should be accompanied by a two-sided confidence interval and one-sided hypothesis tests should be accompanied by one-sided confidence intervals. This will make the \( 1 - \alpha \) level confidence interval the region in which an observation would not lead to rejection of the null hypothesis. A confidence interval should, however, provide more than a rejection region; it should provide an interval in which we can have confidence that the true parameter lies. This corresponds to the interval which provides most support for the parameter. As such confidence intervals should be two-sided even if the appropriate test may be one-sided (Boyles, 2008). We will use two-sided confidence intervals throughout and use these in conjunction with \( p \)-values from one-sided difference and similarity tests. This is also implemented in \texttt{sensR}.

Confidence intervals may, however, be one-sided in a slightly different respect since it may happen, for instance, that the lower confidence limit is at the guessing probability, \( p_g \). If the observed proportion of correct answers is less than \( p_g \), the lower confidence limit will also be higher than the observed proportion.

Confidence intervals may be degenerate in the sense that both limits can be zero; this is obviously not very informative. This may happen if, for instance, the observed proportion is below \( p_g \) and \( \alpha \) is large enough. For small enough \( \alpha \), the upper confidence limit for \( d' \) will, however, exceed zero.

Confidence intervals can be used for difference and similarity testing as argued by MacRae (1995) and Carr (1995) when it is enough to know if the alternative hypothesis is rejected or not. Comparing the formulas for the ‘exact’ Clopper-Pearson confidence limits (23) with
the formulas for \( p \)-values in difference and similarity tests also based on the exact test, it is clear that there is a close connection.

If \( p_{c0} \) under \( H_0 \) is below the lower confidence limit in a \( 1 - \alpha \) level interval, then the \( p \)-value of a difference test will be below \( \alpha/2 \), i.e. the test will be significant at the \( \alpha/2 \)-level. Thus, if \( p_{c0} \) is below the lower confidence limit in a 90\% interval, then the difference test is significant at the 5\% level. Similarly, if \( p_{c0} \) is above the upper confidence limit in a 90\% interval, then the similarity test is significant at the 5\% level.

In difference testing the binomial test is not too liberal even if there is variability in \( p_d \) under the alternative hypothesis, because there can be no variability under the null hypothesis that \( p_d = 0 \). In similarity testing, however, \( p_d > 0 \) under \( H_0 \) and the standard binomial test could possibly be liberal. Also not that \( p_d \) under \( H_A \) will be less than \( p_d \) under \( H_0 \), and if there is variation in \( p_d \) in the distribution, this variation could be larger under \( H_0 \) than under \( H_A \). Also, the power and sample size computations in the following assume that zero variability in \( p_d \). Possibly the power will be lower and sample sizes higher if there really is variation in \( p_d \) in the population.

The similarity tests discussed so far are targeted toward equivalence in the population on average. There is no consideration of equivalence on the level of individual discrimination.

A general problem with discrimination testing outlined so far is the assumption that all assessors have the same probability of discrimination. This is hardly ever a priori plausible. The so-called guessing model (refs) assumes that there are two kinds of assessors; non-discriminators that always guess and true discriminators that always perceive the difference and discriminate correctly. This assumption is also hardly ever a priori plausible. More plausible is perhaps that the probability of discrimination has some distribution across the population of assessors as is assumed in the chance-corrected beta-binomial distribution.

### 3.1.7 Implementation in sensR

The function \texttt{rescale} that was described in section 3.0.1 has an additional optional argument \texttt{std.err} which allows one to get the standard error of, say, \( p_d \) and \( d' \) if the standard error of \( p_c \) is supplied. This is done through application of eq. (16) and (17) and by using the user visible function \texttt{psyderiv}, which implements the derivative of the psychometric functions, \( f'_p(d') \) for the four common discrimination protocols:

\[
\texttt{> rescale(pd = 0.2, std.err = 0.12, method = "triangle"})
\]

Estimates for the triangle protocol:

\[
\begin{array}{c|c|c}
\text{pc} & \text{pd} & \text{d.prime} \\
\hline
1 & 0.4666667 & 0.2 \\
\end{array}
\]

Standard errors:

\[
\begin{array}{c|c|c}
\text{pc} & \text{pd} & \text{d.prime} \\
\hline
1 & 0.08 & 0.12 \\
\end{array}
\]

The \texttt{discrim} function is the primary function for inference in the duo-trio, triangle, 2-AFC and 3-AFC protocols. Given the number of correct answers, \( x \) and the number of trials, \( n \), \texttt{discrim} will provide estimates, standard errors and confidence intervals on the scale of \( p_c \), \( p_d \) and \( d' \). It will also report the \( p \)-value from a difference or similarity test of the users choice. \( p \)-values will be one-sided while confidence limits will be two-sided, cf. section 3.1.6. Confidence intervals are computed on the scale of \( p_c \) and then transformed to the \( p_d \) and \( d' \).
scales as discussed in section 3.1.3. The user can choose between several statistics including the 'exact' binomial, likelihood, Wald and score statistics. The score option leads to the so-called Wilson or score interval, while the \( p \)-value is based on the \( w^* \) statistic, cf. eq. (24).

Estimates and confidence intervals reported by `discrim` respect the allowed range of the parameters, cf. eq. (13) and standard errors are not reported if the parameter estimates are on the boundary of their parameter space (allowed range).

Strictly speaking the Wald statistic (20) is not defined when \( x/n \leq p_g \), since the standard error of \( \hat{p}_c \) is not defined. However, it makes sense to use \( \sqrt{x/n (1-x/n)} \) as standard error in this case. This is adopted in `discrim`.

Similarity testing does not make sense if \( p_{d0} = 0 \) under the null hypothesis, cf. eq. (30), so a positive \( p_{d0} \) has to be chosen for similarity testing.

**Example:** Suppose we have performed a 3-AFC discrimination test and observed 10 correct answers in 15 trials. We want estimates of the \( p_c \), \( p_d \) and \( d' \), their standard error and 95\% confidence intervals. We are also interested in the difference test of no difference and decide to use the likelihood root statistic for confidence intervals and tests. Using the `discrim` function in R we obtain:

```r
> discrim(10, 15, method = "threeAFC", statistic = "likelihood")
```

Estimates for the threeAFC discrimination protocol with 10 correct answers in 15 trials. \( p \)-value and 95\% confidence intervals are based on the likelihood root statistic.

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>pc</td>
<td>0.6666667</td>
<td>0.1217161</td>
<td>0.4154537</td>
</tr>
<tr>
<td>pd</td>
<td>0.5000000</td>
<td>0.1825742</td>
<td>0.1231806</td>
</tr>
<tr>
<td>d-prime</td>
<td>1.1159023</td>
<td>0.4359153</td>
<td>0.2802776</td>
</tr>
</tbody>
</table>

Result of difference test:

likelihood root statistic = 2.632769 \( p \)-value = 0.0042346

Alternative hypothesis: \( d' \) is greater than 0

If instead we had observed 4 correct answers in 15 trials and were interested in the similarity test with \( p_{d0} = 1/5 \) under the null hypothesis, we get using the 'exact' binomial criterion for confidence intervals and tests:

```r
> discrim(4, 15, method = "threeAFC", test = "similarity", pd0 = 0.2, statistic = "exact")
```

Estimates for the threeAFC discrimination protocol with 4 correct answers in 15 trials. \( p \)-value and 95\% confidence intervals are based on the 'exact' binomial test.

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>pc</td>
<td>0.3333333</td>
<td>NA</td>
<td>0.3333333</td>
</tr>
<tr>
<td>pd</td>
<td>0.0000000</td>
<td>NA</td>
<td>0.3265049</td>
</tr>
<tr>
<td>d-prime</td>
<td>0.0000000</td>
<td>NA</td>
<td>0.7227091</td>
</tr>
</tbody>
</table>

Result of similarity test:

'exact' binomial test: \( p \)-value = 0.096376
Figure 2: Relative likelihood function for a 3-AFC experiment with 10 correct answers in 15 trials. The maximum likelihood estimate is at \( d' = 1.12 \) and the two horizontal lines determine the 95% and 99% likelihood based confidence intervals.

Alternative hypothesis: \( d \)-prime is less than 0.4482213

A few auxiliary methods for \texttt{discrim} objects are available. \texttt{confint} returns the confidence intervals computed in the \texttt{discrim} object, \texttt{profile} extracts the (profile) likelihood function and \texttt{plot.profile} plots the likelihood function.

**Example**  To illustrate the auxiliary methods consider the 3-AFC example above where 10 correct answers were observed in 15 trials.

```r
> fm1 <- discrim(10, 15, method = "threeAFC", statistic = "exact")
> confint(fm1)

<table>
<thead>
<tr>
<th></th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>pc</td>
<td>0.3838037</td>
<td>0.8817589</td>
</tr>
<tr>
<td>pd</td>
<td>0.0757056</td>
<td>0.6226583</td>
</tr>
<tr>
<td>d-prime</td>
<td>0.1744201</td>
<td>2.1015496</td>
</tr>
</tbody>
</table>

attr(,"method")
[1] "threeAFC"
attr(,"conf.level")
[1] 0.95
attr(,"statistic")
[1] "exact"

> plot(profile(fm1))
```

The resulting graph is shown in Fig. 2. Observe that the likelihood (profile) function may be extracted from a \texttt{discrim} object that is not fitted with \texttt{statistic = "likelihood"}. Further information about the use and interpretation of (profile) likelihood curves in sensory experiments is given in (Brockhoff and Christensen, 2010; Christensen and Brockhoff, 2009).
3.2 Sample size and power calculations for simple discrimination protocols

The power of a test is the probability of getting a significant result for a particular test given data, significance level and a particular difference. In other words, it is the probability of observing a difference that is actually there. Power and sample size calculations require that a model under the null hypothesis and a model under the alternative hypothesis are decided upon. The null model is often implied by the null hypothesis and is used to calculate the critical value. The alternative model has to lie under the alternative hypothesis and involves a subject matter choice. Power is then calculated for that particular choice of alternative model.

In the following we will consider calculation of power and sample size based directly on the binomial distribution. Later we will consider calculations based on a normal approximation and based on simulations.

3.2.1 The critical value

Formally the critical value, $x_c$, of a one-sided binomial test where the alternative hypothesis is difference, or equivalently greater, is the smallest integer number that satisfies

$$P(X \geq x_c) \leq \alpha \quad \text{where} \quad X \sim \text{binom}(p_c, n) \quad (33)$$

and $p_c$ is the probability of a correct answer under the null hypothesis. Similarly the critical value, $x_c$, of a one-sided binomial test where the alternative hypothesis is similarity, or equivalently less, is the largest integer number that satisfies

$$P(X \leq x_c) \leq \alpha \quad \text{where} \quad X \sim \text{binom}(p_c, n) \quad (34)$$

If the sample size is small for the desired $\alpha$, there may not be a possible critical value that satisfies (33) or (34). In a difference test it may not be enough to observe $x = n$ correct answers, i.e. all correct answers for the test to be significant at the required $\alpha$. Similarly, it may not be enough to observe no correct answers ($x = 0$) for the similarity test to be significant at the required $\alpha$.

A simple way to compute $x_c$ is to use a small while loop (shown here for a difference test):

```plaintext
i = 0
while P(X \geq i) > \alpha do
    i = i + 1
end while
return i + 1
```

However, if $x_c$ is a large number, many iterations of the loop would be required, so instead in the `findcr` function in package `sensR` eq. (33) and (34) are solved numerically for $x_c$. One complication with this method is that $P(X \geq x_c)$ is discontinuous in $x_c$ and that requires special attention.

**Example:** Consider the situation that $X = 15$ correct answers are observed out of $n = 20$ trials in a duo-trio test. The exact binomial $p$-value of a no-difference test is $P(X \geq 15) =$
\[ 1 - P(X \leq 15 - 1) = 0.021, \] where \( X \sim \text{binom}(0.5, 20) \) so this is significant. If on the other hand we had observed \( X = 14 \), then the p-value would have been \( P(X \geq 14) = 0.058 \), which is not significant. We say that \( x_c = 15 \) is the critical value for this particular test on the \( \alpha = 5\% \) significance level because \( x_c = 15 \) is the smallest number of correct answers that renders the test significant.

In R we can find the p-values with

\[
\begin{align*}
> & 1 - \text{pbinom}(q = 15 - 1, \text{size} = 20, \text{prob} = 0.5) \\
[1] & 0.02069473 \\
> & 1 - \text{pbinom}(q = 14 - 1, \text{size} = 20, \text{prob} = 0.5) \\
[1] & 0.05765915
\end{align*}
\]

The while loop looks like

\[
\begin{align*}
> & i <- 0 \\
> & \text{while} \ (1 - \text{pbinom}(q = i, \text{size} = 20, \text{prob} = 0.5) > 0.05) \\
& \quad \{ \\
& \quad \quad i <- i + 1 \\
& \quad \} \\
> & i + 1 \\
[1] & 15
\end{align*}
\]

while we could also use the \texttt{findcr} function in package sensR:

\[
\begin{align*}
> & \text{findcr}(\text{sample.size} = 20, \alpha = 0.05, \text{p0} = 0.5) \\
[1] & 15
\end{align*}
\]

### 3.2.2 The power of difference tests

The power of a difference test is

\[
\text{power} = P(X \geq x_c) \quad \text{where} \quad X \sim \text{binom}(p_{cA}, n), \tag{35}
\]

where \( p_{cA} \) is the probability of a correct answer under the alternative hypothesis and \( x_c \) is the critical value of the test, which depends on the probability of a correct answer under the null hypothesis and the significance level, \( \alpha \).

Power increases with the difference between \( p_{c0} \) and \( p_{cA} \), the sample size and \( \alpha \). Power can be computed directly once the critical value, \( p_{cA} \) and \( n \) are known, so the only computational challenge is in the computation of the critical value.

**Example:** The power of the test considered in the previous example is the probability of getting this p-value or one that is smaller. This depends on the actual sensory difference of the objects/the proportion of discriminators. If half the population are discriminators or equivalently if each assessor has a 50\% of correctly discriminating a set of samples, then \( p_c = 1/2 + 1/2p_d = 3/4 \). The power is the probability of observing 15 or more correct answers:

\[
\text{power} = P(X \geq 15) = 1 - P(X \leq 15 - 1) = 0.617 \quad \text{where} \quad X \sim \text{binom}(3/4, 20) \tag{36}
\]
This can be obtained in R with

```r
> 1 - pbinom(q = 15 - 1, size = 20, prob = 3/4)
[1] 0.6171727
```

or directly using the `discrimPwr` function from `sensR`:

```r
> discrimPwr(pdA = 0.5, sample.size = 20, alpha = 0.05, 
pGuess = 1/2)
[1] 0.6171727
```

Observe that `discrimPwr` requires that the effect size under the alternative hypothesis is given in terms of \( p_d \) rather than \( p_c \) or \( d' \). If the effect size under the alternative hypothesis is formulated in terms of \( d' \), then `rescale` can be used to convert from \( d'_A \) to \( p_{dA} \), but it would be easier to use `d.primePwr`, which accepts \( d'_A \) directly and internally calls `discrimPwr`.

If the significance test of interest is not that of no-difference, but that of a small difference versus a relevant difference, the computation of the critical value is slightly different. The power calculation remain essentially the same. If the limit between irrelevant and relevant differences is at \( p_{dA} = 0.1 \), so \( p_c = 1/2 + 1/2 \cdot 0.1 = 0.55 \), then \( P(X \geq 16 | p_{d0} = 0.55, n = 20) = 1 - P(X \leq 16 - 1) = 0.019 \) while \( P(X \geq 15 | p_{d0} = 0.55, n = 20) = 1 - P(X \leq 15 - 1) = 0.055 \). The critical value is therefore 16 and the power of the test is

\[
\text{power} = P(X \geq 16) = 0.415 \quad \text{where} \quad X \sim \text{binom}(p_{cA} = 3/4, n = 20) \quad (37)
\]

In R we could get the power of this test with

```r
> discrimPwr(pdA = 0.5, pd0 = 0.1, sample.size = 20, alpha = 0.05, 
pGuess = 1/2)
[1] 0.4148415
```

Note the `pd0` argument which should match the value of \( p_d \) under the null hypothesis.

### 3.2.3 The power of similarity tests

The power of a similarity test is

\[
\text{power} = P(X \leq x_c) \quad \text{where} \quad X \sim \text{binom}(p_{cA}, n) \quad (38)
\]

and \( p_{cA} \) is the probability of a correct answer under the alternative hypothesis and \( x_c \) is the critical value of the test, which depends on the probability of a correct answer under the null hypothesis and the significance level, \( \alpha \).

**Example:** Assume that we want to calculate the power of a similarity test using the duo-trio protocol with \( n = 100 \), and that we want to show that the probability of discrimination is less than 1/3, while we believe that there is actually no difference between the objects, so the true probability of discrimination is zero. The null hypothesis is therefore \( H_0 : p_c \geq 1/2 + 1/2 \cdot 1/3 = 2/3 \) and the alternative hypothesis is \( H_A : p_c < 2/3 \). The critical value
of this test is $x_c = 58$ since $p = P(X \leq 58|p_c = 2/3, n = 100) = 0.042 \leq 0.05$ while $P(X \leq 59) = 0.064 > 0.05$. The power of this test is therefore

$$\text{power} = P(X \leq 58|p_c = 0.5, n = 100) = 0.956 \quad (39)$$

We would compute this power in R with

```r
> discrimPwr(pdA = 0, pd0 = 1/3, sample.size = 100, alpha = 0.05, 
  pGuess = 1/2, test = "similarity")
```

[1] 0.955687

If in fact there is a small difference between the objects, so that there is a positive probability of discrimination, say $p_d = 1/5$, then the power is (the critical value remains the same):

$$\text{power} = P(X \leq 58|p_c = 0.5(1 + 1/5), n = 100) = 0.377 \quad (40)$$

We would compute this power in R with

```r
> discrimPwr(pdA = 1/5, pd0 = 1/3, sample.size = 100, alpha = 0.05, 
  pGuess = 1/2, test = "similarity")
```

[1] 0.3774673

Observe how the power of the similarity test is quite good if there is absolutely no observable difference between the objects, while if there is in fact a small probability that a difference can be observed, the power is horrible and the sample size far from sufficient.

### 3.2.4 Power calculation based on simulations

In more complicated models it is not possible to determine an explicit expression for the power of a test and calculation of power based on simulations can be an attractive approach. Sometimes it may also just be easier to let the computer do the job by running simulations rather than to get bugged down in derivations of explicit expressions for power even though they may in fact be possible to derive.

Recall that power is the probability of getting a significant result when there is in fact a difference, thus in the long run it is the proportion of significant results to the total number of tests:

$$\text{power} = \frac{\text{no. } p\text{-values} < \alpha}{\text{no. tests}} \quad (41)$$

We can let the computer generate random data from the model under the alternative hypothesis and then perform the significance test. We can even do that many many times and record the $p$-values allowing us to calculate the power via eq. (41). In the following we will do exactly that for a binomial test for which we know the right answer.

Consider the no-difference example above in section 3.2.2 where $n = 20$ and the power of a no-difference test was 0.617 when $p_d = 1/2$, so $p_c = 3/4$. We will estimate the power via simulation by generating 10,000 (pseudo) random draws, $X_i$, $i = 1, \ldots, 10,000$ from $X_i \sim \text{binom}(p_c = 3/4, n = 20)$. For each of these draws we calculate the $p$-value as $p_i = P(X \geq x_i|p_c = 1/2, n = 20)$. Among these $p$-values 6184 were below 0.05, so the power estimated by simulation is 0.6184. Observe that this is close to, but not exactly the power that we obtained analytically (0.617). If we did the power calculation over again, we would
most likely get a slightly different power estimate although probably also close to 0.617 because we would obtain a slightly different set of random draws. This illustrates that although power calculation via simulation is simple, the result varies a little from one run to another.

Fortunately we can estimate the uncertainty in the estimated power from standard binomial principles. The standard error of the estimated power is $\text{se}(\hat{\text{power}}) = \sqrt{\text{power}(1 - \text{power})/n_{\text{sim}}} = \sqrt{0.6814(1 - 0.6814)/10,000} = 0.0049$ and an approximate Wald 95% CI for the estimated power is $[0.609; 0.628]$, which covers the true value (0.617) as one would expect.

3.2.5 Power calculation based on the normal approximation

An often used approximation for power and sample size calculations is the normal approximation; the idea is to use a statistic that asymptotically follows a standard normal distribution. For a binomial parameter power and sample size calculation may be based on the Wald statistic (20) as for example described by Lachin (1981) and advocated by Bi (2006) in a sensometric context. We are not aware of any numerical assessments of the accuracy of the normal approximation for power and sample size calculations, but we may expect that for small $n$ or $p$ (under the null or alternative) close to one, the approximation may be rather inaccurate. Since power and sample size determinations are readily available for the exact binomial test, we see no reason to use approximate statistics with doubtful properties for these purposes.

Consider the following hypotheses for a binomial parameter:

$$H_0 : p = p_0 \quad H_A : p > p_0,$$

then under the null hypothesis approximately

$$\frac{\hat{p} - p_0}{\sigma_0} \sim N(0, 1)$$

and under the alternative hypothesis approximately

$$\frac{\hat{p} - p_A}{\sigma_A} \sim N(0, 1),$$

where $p_A$ is the probability under the alternative hypothesis, $\sigma_0 = \sqrt{p_0(1 - p_0)/n}$, $\sigma_A = \sqrt{p_A(1 - p_A)/n}$ and $\hat{p} = X/n$ is the estimator of a binomial parameter. The critical point above which the null hypothesis is rejected is then

$$\frac{\hat{p} - p_0}{\sigma_0} > \Phi^{-1}(1 - \alpha) = z_{1-\alpha}$$

i.e. when

$$\hat{p} > z_{1-\alpha} \sigma_0 + p_0.$$

Under $H_A$ the null hypothesis is rejected if

$$\frac{\hat{p} - p_A}{\sigma_A} \geq \frac{z_{1-\alpha} \sigma_0 + p_0 - p_A}{\sigma_A}$$

and the power is

$$\text{power} = P\left(Z > \frac{z_{1-\alpha} \sigma_0 + p_0 - p_A}{\sigma_A}\right) = 1 - \Phi\left(\frac{z_{1-\alpha} \sigma_0 + p_0 - p_A}{\sigma_A}\right)$$
Equivalent considerations for the equivalence hypotheses lead to

$$\text{power} = P\left(Z < \frac{z_\alpha \sigma_0 + p_0 - p_A}{\sigma_A}\right) = \Phi\left(\frac{z_\alpha \sigma_0 + p_0 - p_A}{\sigma_A}\right)$$ \hspace{1cm} (49)$$

Isolating $n$ in eq. (48) leads to the following expression for the sample size of difference tests:

$$\text{sample size} = \left(\frac{z_\beta \sqrt{p_A(1-p_A)} - z_{1-\alpha} \sqrt{p_0(1-p_0)}}{p_0 - p_A}\right)^2,$$ \hspace{1cm} (50)

where $z_\beta = \Phi^{-1}(1 - \text{power})$. Equivalently for similarity tests:

$$\text{sample size} = \left(\frac{z_{1-\beta} \sqrt{p_A(1-p_A)} - z_\alpha \sqrt{p_0(1-p_0)}}{p_0 - p_A}\right)^2,$$ \hspace{1cm} (51)

where $z_{1-\beta} = \Phi^{-1}(\text{power})$. The sample sizes given by (50) and (51) should be rounded up to the nearest integer.

### 3.2.6 Sample size determination

In principle sample size determination is simple; find the sample size such that the power is sufficiently high for a particular test at some significance level given some true difference. Computationally, however, it can be a challenge.

Formally, the required sample size, $n^*$ for a sensory difference test is the smallest integer number, $n^*$ that satisfies

$$P(X \geq x_c) \geq \text{target-power} \quad \text{where} \quad X \sim \text{binom}(p_c, n^*),$$ \hspace{1cm} (52)

and $P(X \geq x_c)$ is the actual power of the test. Power for a difference test only increases with increasing sample size if the true difference, $p_d$ is larger than the null difference, $p_{d0}$, so it is a requirement that the value of $p_d$ specified as the true difference is actually covered by the alternative hypothesis.

Similarly, the required sample size, $n^*$ for a similarity test is the smallest integer number, $n^*$ that satisfies

$$P(X \leq x_s) \geq \text{target-power} \quad \text{where} \quad X \sim \text{binom}(p_c, n^*),$$ \hspace{1cm} (53)

and $P(X \leq x_s)$ is the actual power of the test. Power only increases with increasing sample size if the true difference, $p_d$ is less than the null difference, $p_{d0}$, so as for difference tests, the value specified as the true difference has to be covered by the alternative hypothesis.

The sample size depends on the particulars of the null and alternative hypotheses as well as the significance level of the test, i.e. $\alpha$ and the desired minimum power; the target-power.

So much for the formal definitions: practical sample size determination is in fact not as simple as the definitions may lead one to believe. Consider a situation in which we want to know which sample size to choose in a difference test using the triangle protocol where the null hypothesis is no difference, target power is 0.80, and we believe the actual difference is $d' = 0.9$ under the alternative hypothesis. Standard sample size calculations under the
Figure 3: The relation between sample size and power for a difference test with the triangle protocol. The null hypothesis is that of no difference and $d' = 0.9$ is assumed under the alternative model. A power of 0.80 is desired.

definition (52) tells us that 297 tests are enough; this leads to an actual power of 0.802. However, had we decided to use, say, 300 tests—for convenience and just to be on the safe side, the power of the test is only 0.774; much less than the power with 297 tests and below our target power. This is truly worrying; how many samples do we need to be sure that all larger sample sizes also lead to a power above 0.80? It is natural to expect power to increase with every increase in sample size (a monotonic increase in power with sample size), but this is not the case as is illustrated in Fig. 3.

Power generally increases with the sample size, but it does so in a zig-zag way due to the discreteness of the binomial distribution. As is seen in Fig. 3, the smallest sample size for which power is higher than 0.80 is 297 (actual power = 0.802). The next sample size that gives a power above 0.80 is 302, but the actual power is now less than 0.801. We would need 305 samples (actual power = 0.806) to obtain a power that is higher than the power with 297, and no less than 318 samples (actual power = 0.802) if no larger sample size should lead to a power less than 0.80.

Even though an increase in sample size may lead to a decrease in power, it will instead lead to a decrease in the actual $\alpha$-level. This occurs because the critical value of the test is at times piece-wise constant as a function of sample size, cf. Fig. 4.

The sample size for the exact binomial test may be computed with much the same while loop that could also be used to find the critical value (cf. section 3.2.1):

$\begin{align*}
  \text{i} &= 1 \\
  \text{\textbf{while}} & \text{ actual power(i) < target power} \text{ \textbf{do}} \\
  \text{i} &= \text{i} + 1
\end{align*}$
where actual power depends on the hypothesis, cf. (52) and (53). The problem with this approach is that if the required sample size is large, it may take some time to get there; recall that at every evaluation of the actual power, the critical value has to be determined. Due to the non-monotonicity of the relationship between power and sample size (cf. Fig. 3), it is not possible to simply solve for the required sample size numerically.

An improvement over the simple while loop is suggested by the normal approximation to the required sample size shown in Fig. 3 in blue. This approximation seems to estimate the sample size too low, and to do so consistently. For the example considered here, the normal approximation estimates that 291 samples is enough to obtain a power of 0.80 (actual power = 0.8001). The while loop could simply be started at $i = 291$ rather than at $i = 1$. A problem with this approach is that the normal approximation is not always strictly liberal. In the function `discrimSS` in package `sensR` a compromise is used, where the while loop is started at one if the sample size estimated by the normal approximation is less than 50. Otherwise the while loop is started at 90% of the normal approximation estimate and sometimes even lower if necessary. If the normal approximation estimate is larger than 10,000, the function will inform of that and not attempt to estimate the sample size due to the expected large computation time. In addition to the sample size for the ‘exact’ binomial test, it is also possible to ask for the sample size based on the normal approximation.

**Example:** Consider the example above illustrated in Fig. 3; we wanted to know the sample size for a difference test where the null hypothesis is that of no difference using the triangle protocol. We want a power of 0.80, take $\alpha = 0.05$ and we assume the actual difference is $d' = 0.9$ under the alternative hypothesis. Using package `sensR` we may get the sample size for the exact binomial test with

```r
> (pd <- coef(rescale(d.prime = .9, method = "triangle"))$pd)
[1] 0.1044969

> discrimSS(pdA = pd, pd0 = 0, target.power = 0.8, alpha = 0.05, pGuess = 1/3, test = "difference", statistic = "exact")

[1] 297
```

Figure 4: Left: critical value of the test. Right: actual $\alpha$-level of the test.
We could also obtain the normal approximation with

```r
> discrimSS(pdA = pd, pd0 = 0, target.power = 0.8, alpha = 0.05, pGuess = 1/3, test = "difference", statistic = "normal")
```

[1] 291

References


Clopper, C. J. and E. S. Pearson (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika 26, 404–413.


MacRae, A. W. (1995). Confidence intervals for the triangle test can give assurance that products are similar. Food Quality and Preference 6(61-67).


