Quality Assessment of Adaptive Bitrate Videos using Image Metrics and Machine Learning

Søgaard, Jacob; Forchhammer, Søren; Brunnström, Kjell

Published in:
Proceedings of Qomex 2015

Link to article, DOI:
10.1109/qomex.2015.7148105

Publication date:
2015

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Quality Assessment of Adaptive Bitrate Videos using Image Metrics and Machine Learning

Jacob Søgaard¹, Søren Forchhammer¹, and Kjell Brunnström²,³
¹ Technical University of Denmark, Kgs Lyngby, Denmark
² Acreo Swedish ICT AB, Kista, Sweden
³ Mid Sweden University, Sundsvall, Sweden

Abstract—Adaptive bitrate (ABR) streaming is widely used for distribution of videos over the internet. In this work, we investigate how well we can predict the quality of such videos using well-known image metrics, information about the bitrate levels, and a relatively simple machine learning method. Quality assessment of ABR videos is a hard problem, but our initial results are promising. We obtain a Spearman rank order correlation of 0.88 using content-independent cross-validation.

I. INTRODUCTION

In recent years the amount of video traffic over the internet has grown. HTTP Adaptive Streaming (HAS) is a popular method for delivering video over the internet. It adapts the video to the current network conditions, server load and end user device capabilities. Video Quality Assessment (VQA) is an important tool to ensure the Quality of Experience (QoE) for video delivery. QoE for ABR videos is not well understood and by extension VQA of ABR videos is a hard problem [1]. In this work we consider the effects of ABR as reflected in the LIVE mobile database [2], apart from frame freeze. We present some initial results of an early-stage VQA tool for ABR videos. The approach in this paper is to use well-known image quality metrics to assess the frames of a video and using different pooling methods, calculate several features for the video. Additionally we use simple information about the bitrate levels of the video i.e. the number of increasing and decreasing steps in the video. Finally, we use a relatively simple machine learning method to map the features to a quality score for the video. For related work see e.g. [3], [4].

II. IMAGE METRICS AND VIDEO FEATURES

We use three different image metrics to assess the quality of the individual frames in a video: PSNR, SSIM [5], and MS-SSIM [6]. For SSIM and MS-SSIM we use the default parameters. To define quality-relevant features for the video we use temporal pooling on the image metrics scores over the frames as described in this section. The objective image metrics are used separately.

As a measure of the average quality and the variation hereof, we calculate the mean \( \mu \) and standard deviation \( \sigma \) of the objective metric scores. Since the perception of quality also depends on recency [1], we also calculate the mean of the metric scores corresponding to the first and last 2 seconds of the video, respectively. Finally, inspired by [7] we divide the objective scores into clusters depending on the image metric scores using the method from [8], such that we obtain clusters of varying quality levels. Then a weighted average of the objective score from the clusters with lowest \( \mu_L \) and highest means \( \mu_H \) are calculated as:

\[
\tilde{\mu} = \frac{\sum_{i \in C_L} S_i + w \sum_{i \in C_H} S_i}{|C_L| + w|C_H|}
\]

(1)

where \( S_i \) are the subjective scores indexed by \( i \), \( C_L \) and \( C_H \) are the set of clusters with lowest and highest mean, respectively. \(| \cdot |\) denotes the size of each set. The weight \( w \) is defined as:

\[
w = \left( \frac{\mu_L}{\mu_H} \right)^2
\]

(2)

In this way, the low objective scores will carry more weight in sequences where there is a large difference between good and bad quality. The weight \( w \) is also used as a video feature, since it carries information about the difference in quality levels in the video. Thus, a total of 6 features based on the image metric is produced. Note, that in order to ensure the features to be in the interval \([0; 1]\) for numerical reasons, we rescale the PSNR-values to this interval using predetermined threshold values for minimum and maximum PSNR. Additionally, 2 features based on the bitrate levels in the video are also produced. They are simply defined as the number of increasing steps and decreasing steps of bitrate levels per second. The proposed Full-Reference (FR) model is thus defined by these 8 features and the mapping outlined in Section III.

III. MACHINE LEARNING

We use the method known as the Elastic Net (EN) to map the features to a quality score. The goal is to estimate the coefficients \( \beta \) of a regularized linear regression model:

\[
\tilde{\beta} = \arg\min_{\beta} ||y - X\beta||^2 + \lambda_2 ||\beta||^2 + \lambda_1 ||\beta||_1
\]

(3)

where \( y \) is the target quality values, \( X \) is a feature matrix with rows of feature vectors, and \( \lambda_1 \) and \( \lambda_2 \) are regularization parameters of the \( L^1 \)-norm and the \( L^2 \)-norm, respectively. Due to the \( L^1 \)-norm in (3) the solution of an EN can generally be considered to be sparse and therefore feature selection is inherently a part of the EN method. For more information about the EN method we refer to [9]. In our experiments we use the implementation of the EN presented in [10].

IV. RESULTS

To test the performance of our methods we use the LIVE mobile database [2] (which consists of 200 distorted videos). Since we do not consider freezing in our assessment, videos with this kind of artifact is removed from the dataset (in
Table I. Cross-validation performance.

<table>
<thead>
<tr>
<th></th>
<th>VQ - PSNR</th>
<th>VQ - SSIM</th>
<th>VQ - MS-SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\bar{x} )</td>
<td>(\mu )</td>
<td>(\sigma )</td>
</tr>
<tr>
<td>SROCC</td>
<td>0.88</td>
<td>0.83</td>
<td>0.11</td>
</tr>
<tr>
<td>LCC</td>
<td>0.87</td>
<td>0.85</td>
<td>0.075</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.57</td>
<td>0.61</td>
<td>0.14</td>
</tr>
<tr>
<td>OR</td>
<td>0.41</td>
<td>0.40</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Table II. Training performance.

<table>
<thead>
<tr>
<th></th>
<th>VQ - PSNR</th>
<th>VQ - SSIM</th>
<th>VQ - MS-SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\bar{x} )</td>
<td>(\mu )</td>
<td>(\sigma )</td>
</tr>
<tr>
<td>SROCC</td>
<td>0.82</td>
<td>0.80</td>
<td>0.80</td>
</tr>
<tr>
<td>LCC</td>
<td>0.81</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.54</td>
<td>0.60</td>
<td>0.60</td>
</tr>
<tr>
<td>OR</td>
<td>0.33</td>
<td>0.43</td>
<td>0.44</td>
</tr>
</tbody>
</table>

In this paper we have developed a FR model for predicting the quality of video in an ABR setting. Though the model is still in early-stage, we get promising results with a cross-validation correlation up to 0.88. There is still room for improvement such as adding more sophisticated video features and taking freezing events into account. Using our machine learning approach, we can also evaluate some features as less relevant, such as the mean objective quality score of the first 2 seconds of a video. It would be interesting to consider how the knowledge gained from this work could be used in building a NR model for the ABR scenario. Lastly, as future work our method should be tested on an independent dataset, to measure the test error performance in this case.

REFERENCES


