Compressive Online Robust Principal Component Analysis with Multiple Prior Information

Van Luong, Huynh; Deligiannis, Nikos; Seiler, Jürgen; Kaup, Andre; Forchhammer, Søren

Publication date:
2017

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

Citation (APA):
1. Motivation

- Applications: Computer vision, web data analysis, anomaly detection, and data visualization, etc.
- Robust Principal Component Analysis (RPCA): Batch-based, decomposes all data samples (matrix M) into low-rank (L) and sparse (S), e.g., all frames in a video, high computational and memory requirements

\[
\min_{L,S} \| L \|_1 + \lambda \| S \|_1 \quad \text{subject to} \quad M = L + S
\]

Challenges
- Online method processing a sequence of signals per time instance from a small set of measurements
  \[ y_l = \Phi(x_l + v_l) \]
  \[ M_l = L_l + S_l \quad \text{into} \quad S_l = [x_l \ldots x_l], \quad L_l = [v_l \ldots v_l] \]
- Minimization at time instance \( l \)

\[
\min_{L_l,S_l} \| L_l \|_1 + \lambda \| S_l \|_1 \quad \text{subject to} \quad y_l = \Phi(x_l + v_l)
\]

where \( \lambda > 0 \) and \( \lambda > 0 \) are weights across the side information signals, and \( W_l \) is a diagonal matrix with weights for each element in the side information signal \( x_l \); namely, \( W_l = \begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & w_{n,n} \end{bmatrix} \) with \( w_{i,i} > 0 \) being the weight for the \( i \)-th element in the \( x_l \) vector.

2. Compressive Online RPCA (CORPCA) With Multiple Prior Information

**Problem formulation**
- Incorporating multiple prior information: at time instance \( l \)
  - we observe \( y_l = \Phi(x_l + v_l) \) with \( y_l \in \mathbb{R}^n \) given priors \( x_l \) and \( v_l \) from \( \{x_l, \ldots, x_{l-1}\} \) and \( \{v_l, \ldots, v_{l-1}\} \)

\[
\min_{x_l,v_l} \| L_l \|_1 + \lambda \| S_l \|_1, \quad \text{subject to} \quad M_l = L_l + S_l
\]

\[
\min_{x_l,v_l} \| L_l \|_1 + \lambda \| S_l \|_1 + \frac{1}{2} \| \Phi(x_l + v_l) - y_l \|_2^2
\]

The CORPCA algorithm
- Solving \( \Phi \)-linear minimization via the soft thresholding operator and the single value thresholding operator, at iteration \( k + 1 \)

\[
\begin{align*}
\mathbf{x}^{k+1} &= \text{arg min} \left\{ \beta \mathbf{h}(\mathbf{x}) : \| \mathbf{x} - \frac{1}{2} \mathbf{u}_0^T \mathbf{y} \|_2^2 \right\} \\
\mathbf{v}^{k+1} &= \text{arg min} \left\{ \alpha \mathbf{h}(\mathbf{v}) : \| \mathbf{v} - \frac{1}{2} \mathbf{u}_0^T \mathbf{y} \|_2^2 \right\}
\end{align*}
\]

where \( \beta, \alpha > 1 \) (superiority of \( \Phi \)-linear minimization over compressive measurements) and \( \mathbf{h} \) is the soft thresholding function.

- Updating weights \( \beta, \alpha \) and \( \mathbf{W}_l \)

3. Experimental Results

**Synthetic data**
- Generating low-rank components: \( n = 500, \ d = 100 \) (training), \( n = 100 \) (testing), \( r = 5 \) (rank)

\[
L = UV^T, \quad \text{where} \quad U \in \mathbb{R}^{n \times r} \quad \text{and} \quad V \in \mathbb{R}^{r \times r}
\]

yields \( L = [v_1 \ldots v_r] \)
- Generating sparse components with \( \| x_l \|_0 = 50 \)

\[
|x_l|_0 = 50 \quad \text{and} \quad |x_l|_0 = 100
\]

- Obtaining \( S = [x_1 \ldots x_r] \)

\[
S = [x_1 \ldots x_r]
\]

- Testing on \( M = [x_1 + v_1 \ldots x_r + v_r] + \mathbf{W}_l \)

\[
M = [x_1 + v_1 \ldots x_r + v_r] + \mathbf{W}_l
\]
- Measuring probabilities of successful decomposition, \( P(\text{success}) \), success \( \iff \| x_l - x_l \|_2 / \| x_l \|_2 \leq 10^{-2} \)

**Compressive video foreground-background separation**
- Considering two videos, Bootstrap (60x80 pixels) and Curtain (64x80 pixels) having a static and a dynamic background, respectively
- Background-foreground video separation with full access to the video data

\[
\text{Compressive separating by varying rates on the number of measurements} \quad m \text{ over the dimension of the data} \quad n
\]

4. Summary

**Solution for an \( \lambda \)-linear minimization**
- Incorporating efficiently multiple prior information
- Updating iteratively weights

The proposed COPRCA algorithm
- Processing a data vector per time instance using compressive measurements
- Solving the \( \lambda \)-linear minimization and updating priors for the next instance

Evaluation of COPRCA on synthetic data and actual video data
- Outperforming classical compressive sensing (CS) \( l_1 \) minimization and CS with single prior information \( l_1 \)-minimization
- The superior performance improvement compared to the existing methods

COPRCA source code, test sequences, and the corresponding outcomes.
[Available: https://github.com/huyenthink/coprca]