Signal Processing 100 (2014) 124–131

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast communication

Iterative partial matrix shrinkage algorithm for matrix rank minimization

Katsumi Konishi [a,*], Kazunori Uruma [b], Tomohiro Takahashi [b], Toshihiro Furukawa [b]

[a] 1-24-2 Nishi-Shinjuku, Shinjuku-ku, Tokyo 163-867, Japan
[b] 1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan

ARTICLE INFO

Article history:
Received 29 June 2013
Received in revised form
9 December 2013
Accepted 16 January 2014
Available online 24 January 2014

Keywords:
Matrix rank minimization
Compressed sensing

ABSTRACT

This paper proposes a new matrix shrinkage algorithm for matrix rank minimization problems. The proposed algorithm provides a low rank solution by estimating a matrix rank and shrinking non-dominant singular values iteratively. We study the convergence properties of the algorithm, which indicate that the algorithm gives approximate low-rank solutions. Numerical results show that the proposed algorithm works efficiently for hard problems with low computing time.

1. Introduction

This paper deals with the following affine rank minimization problem:

$$\text{Minimize } \mathbf{rank}\, X \quad \text{subject to } \mathcal{A}(X) = \boldsymbol{b}, \tag{1}$$

where $X \in \boldsymbol{R}^{m \times n}$ is a design variable, $m \le n$, $\mathcal{A} : \boldsymbol{R}^{m \times n} \to \boldsymbol{R}^p$ is a given linear operator, and $\boldsymbol{b} \in \boldsymbol{R}^p$ is a constant vector. Unfortunately, this problem is NP-hard in general however has a lot of applications in engineering such as collaborative filtering [1], low-order model fitting and system identification [2], image inpainting [3] and quantum state tomography [4]. Therefore it is important to provide a fast algorithm to obtain approximately low-rank solutions.

Several useful and practical algorithms have been proposed [5–14],[1] and most of them use the singular value decomposition (SVD). While the SVD takes a lot of computational cost for large size matrices, we can reduce the cost by utilizing the truncated approximate SVDs such as Linear-Time SVD [15] and randomized SVD [16]. The truncated SVDs require the rank of X, and therefore it is important to estimate the rank. According to rank estimation schemes, the SVD based algorithms can be grouped into two approaches: a high rank approximation approach and a low rank approximation approach. The iterative reweighted least squares (IRLS) algorithm [5] and the fixed point continuation algorithms (FPCA) [8] carry out the SVD of a high rank matrix (usually a full rank matrix) at the first iteration and then lower rank matrices. Since the SVD of high rank matrices is required even when the optimal solution is very low rank, these algorithms take a lot of computing time for large size problems. Contrarily, the singular value thresholding (SVT) algorithm [10], PowerFactorization (PF) [11], ADMiRA [13] and the singular value projection (SVP) algorithm [14] give a candidate for a low-rank solution as very low rank matrix (usually a rank-1 matrix) at the first iteration and then updates it as a higher rank matrix. Since the truncated SVD of very low rank matrices is much faster than that of full rank matrices, these algorithms are fast for large size problems with very low rank solutions. Motivated by this approach, this paper proposes a rank minimization algorithm which begins with a rank-1 approximation and avoids the full SVD. A lot of numerical results show that the IRLS algorithm is one

* Corresponding author.
E-mail address: konishi@kk-lab.jp (K. Konishi).

[1] We can find other useful algorithms at https://sites.google.com/site/igorcarron2/matrixfactorizations/.

0165-1684/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2014.01.014

of the best algorithms that solve harder problems, and therefore the objective of this paper is to propose a new rank minimization algorithm which has the same performance to solve hard problems as IRLS and takes less computing time using a low rank approximation approach.

First this paper provides an iterative partial matrix shrinkage (IPMS) algorithm for the affine rank minimization with given rank, and then a rank estimation scheme which starts with the rank-1 approximation is proposed. Because IPMS empirically recovers a low rank matrix well even when the given rank is smaller than the true rank, IPMS is suitable for a low rank approximation approach, which leads to computational cost reduction by applying the truncated random SVD. Next the convergence properties are provided to show that IPMS gives an approximate low-rank solution. Finally, numerical examples show that IPMS has a good performance to solve the rank minimization problem and takes low computing time comparing with other algorithms.

2. Main results

First we consider the following feasibility problem associated with (1):

$$\text{Find } X \quad \text{subject to } \mathbf{rank}\, X = r, \quad \mathcal{A}(X) = \boldsymbol{b}, \tag{2}$$

where r is a given constant. Let the SVD of X to be given by $X = U\Sigma V^T$, where

$$\Sigma = diag([\sigma_1\ \sigma_2\ \dots\ \sigma_m]^T), \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m, \tag{3}$$

and define a singular value thresholding operator $\mathcal{D}_r : \boldsymbol{R}^{m\times n} \to \boldsymbol{R}^{m\times n}$ as $\mathcal{D}_r(X) = UD_r\Sigma V^T$, where D_r is a diagonal matrix whose first r elements are 0 and whose other elements are 1. Trivially X is a solution of (2) if and only if it satisfies the following equations:

$$\begin{cases} \mathcal{D}_r(X) = \mathbf{0}_{m,n} \\ \mathcal{A}(X) = \boldsymbol{b}. \end{cases} \tag{4}$$

Similar to the constraint removal (CR) algorithm for the sparse optimization [17], this paper provides an algorithm consisting of two interleaved parts in each iteration, estimating the solution X for given D_r and estimating a new D_r. Contrary to the case of the CR algorithm, it is hard to solve Eq. (4), and therefore this paper proposes a matrix shrinkage based algorithm.

In order to describe a scheme to obtain a solution of (4), we define a partial matrix shrinkage operator $\mathcal{T}_{r,\lambda}(X) : \boldsymbol{R}^{m\times n} \to \boldsymbol{R}^{m\times n}$ as follows:

$$\mathcal{T}_{r,\lambda}(X) = U\, diag\left(\begin{bmatrix} \boldsymbol{\sigma}_r \\ \boldsymbol{\sigma}_r^+ \end{bmatrix}\right) V^T,$$

where

$$\boldsymbol{\sigma}_r = [\sigma_1\ \dots\ \sigma_r]^T,$$

$$\boldsymbol{\sigma}_r^+ = [(\sigma_{r+1} - \lambda)_+\ \dots\ (\sigma_m - \lambda)_+]^T,$$

and $(a)_+$ is defined as $(a)_+ = \max(a, 0)$. This paper proposes the following iterative scheme to find X satisfying (4):

$$\begin{cases} Y^{k+1} = \mathcal{T}_{r,\lambda^k}(X^k) \\ X^{k+1} = Y^{k+1} - \mathcal{A}^*(\mathcal{A}(Y^{k+1}) - \boldsymbol{b}), \end{cases} \tag{5}$$

where X^k and λ^k denote a candidate for X and a shrinkage parameter at the kth iteration, respectively. While most matrix shrinkage based algorithms shrink all singular values, the above scheme does not shrink the top r singular values to find a rank r matrix. We can confirm that X^k always satisfies $\mathcal{A}(X^k) = \boldsymbol{b}$ and that $\mathcal{D}_r(Y^k) = \mathbf{0}_{m,n}$ if $\lambda^k \geq \sigma_r$. The update (5) with $\lambda^k = \sigma_r$ is equivalent to the iterative hard thresholding (IHT) algorithm in [9]. Experimental results show that the hard thresholding ($\lambda^k = \sigma_r$) works well for the problem (2), that is, the problem with given rank of X, however, its performance is worse when the given rank is different from the true rank. If λ^k is small, the singular values σ_i for $i > r$ are decreased and converge to 0 gradually, which leads the update (5) to be robust for misestimation of $\mathbf{rank}\, X$. This paper proposes the update rule to determine λ^k as $\lambda^k = \delta\sigma_r^k$, where $\delta \in (0, 1]$, and σ_r^k denotes the rth largest singular value of X^k. Then the iterative partial matrix shrinkage (IPMS) is proposed as shown in Algorithm 1, where $\eta_\delta > 1$, and δ is reduced gradually to achieve fast convergence and precise recovery of a low rank matrix. Comparing with other matrix shrinkage algorithm, this algorithm is more robust for the case that the given rank r is less than the true rank, which can be seen in Section 4.

Algorithm 1. Iterative partial matrix shrinkage (IPMS).

Input: X^0, δ_0, ε, η_δ,
  $k \leftarrow 0$.
  $\delta \leftarrow \delta_0$.
  while not converge **do**
    repeat
      $[U^k, \sigma_1^k, \sigma_2^k, \dots, \sigma_m, V^k] \leftarrow \mathrm{SVD}(X^k)$.
      $r^k \leftarrow$ given rank r or estimated by Algorithm2
      $\lambda^k \leftarrow \delta\sigma_{r^k}^k$.
      $Y^{k+1} \leftarrow \mathcal{T}_{r^k,\lambda^k}(X^k)$.
      $X^{k+1} \leftarrow Y^{k+1} - \mathcal{A}^*(\mathcal{A}(Y^{k+1}) - \boldsymbol{b})$.
      $k \leftarrow k+1$.
    until $\|X^{k+1} - X^k\|_F / \|X^k\|_F < \varepsilon$
    $\delta \leftarrow \delta/\eta_\delta$.
  end while
Output: low-rank solution X^k

Next we focus on the problem of estimating the rank of X. This paper proposes a matrix rank estimation heuristic as shown in Algorithm 2, where $0 < \alpha_{min} < \alpha^0 \leq 1$ and $\eta \geq 1$. We use this algorithm at each iteration in Algorithm 1. The algorithm assumes that the singular values of X tend to separate into two clusters and estimates its rank by using $\alpha\sigma_1^k$ as the threshold of these clusters, that is, the algorithm gives the rank r if $\sigma_r^k \geq \alpha\sigma_1^k > \sigma_{r+1}^k$. If $\eta = 1$, that is, α^k is a constant for all k, this estimation scheme is exactly the same as that of [5]. The value of α is gradually decreased as $\alpha = \alpha_0/(\eta_\alpha)^{k-1}$. In the case of $\alpha^0 = 1$, it estimates the rank of X^k as 1 at the beginning of iterations and then provides gradually higher rank, that is, it usually gives lower rank than the true rank. Since IPMS is empirically robust for the case