Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 528281, 8 pages
http://dx.doi.org/10.1155/2013/528281

Research Article

On the Low-Rank Approximation Arising in the Generalized Karhunen-Loeve Transform

Xue-Feng Duan,[1] **Qing-Wen Wang,**[2] **and Jiao-Fen Li**[1]

[1] *College of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin 541004, China*
[2] *Department of Mathematics, Shanghai University, Shanghai 200444, China*

Correspondence should be addressed to Qing-Wen Wang; wqw858@yahoo.com.cn

Received 11 March 2013; Accepted 25 April 2013

Academic Editor: Masoud Hajarian

We consider the low-rank approximation problem arising in the generalized Karhunen-Loeve transform. A sufficient condition for the existence of a solution is derived, and the analytical expression of the solution is given. A numerical algorithm is proposed to compute the solution. The new algorithm is illustrated by numerical experiments.

1. Introduction

Throughout this paper, we use $R^{m\times n}$ to denote the set of $m\times n$ real matrices. We use A^T and A^+ to denote the transpose and Moore-Penrose generalized inverse of the matrix A, respectively. The symbol $O^{n\times n}$ stands for the set of all $n\times n$ orthogonal matrices. The symbols $\operatorname{rank}(A)$ and $\|A\|_F$ stand for the rank and the Frobenius norm of the matrix A, respectively. For $a=(a_i)\in R^n$, the symbol $\|a\|$ stands for the l_2-norm of the vector a, that is, $\|a\|_2=(\sum_{i=1}^n a_i^2)^{1/2}$. The symbol $A^{1/2}$ stands for the square root of the matrix A, that is, $(A^{1/2})^2=A$. For the random vector $x=(x_i)\in R^n$, we use $E\{x_i\}$ to stand for the expected value of the ith entry x_i, and we use $E\{xx^T\}=(e_{ij})_{n\times n}$ to stand for the covariance matrix of the random vector x, where $e_{ij}=E[(x_i-E\{x_i\})(x_j-E\{x_j\})]$, $i,j=1,2,\ldots,n$.

The generalized Karhunen-Loeve transform is a well-known signal processing technique for data compression and filtering (see [1–4] for more details). A simple description of the generalized Karhunen-Loeve transform is as follows. Given two random vectors $x\in R^n$, $s\in R^m$ and an integer d $(1\le d<\min\{m,n\})$, the generalized Karhunen-Loeve transform is presented by a matrix T^*, which is a solution of the following minimization problem (see [1, 4]):

$$\min_{T\in R^{m\times n},\ \operatorname{rank}(T)=d} E\left\{\|s-Tx\|^2\right\}. \tag{1}$$

Here the vector s depends on some prior knowledge about the data x.

Without the rank constraint on T, the solution of the minimization problem (1) is

$$T_0=R_{sx}R_x^+, \tag{2}$$

where $R_{sx}=E\{sx^T\}$, $R_x=E\{xx^T\}$. The minimization problem with this case is associated with the well-known concept of Wiener filtering (see [3]).

With the rank constraint on T, that is, $\operatorname{rank}(T)=d$, we first consider the cost function of the minimization problem (1). By using the fact $R_{sx}R_xR_x^+=R_{sx}$ and the four Moore-Penrose equations of R_x^+, it is easy to verify that (see also [1])

$$E\left\{\|s-Tx\|^2\right\}=\operatorname{tr}\left\{(T-T_0)R_x(T-T_0)^T\right\}+E\left\{\|s-T_0x\|^2\right\}. \tag{3}$$

Noting that the covariance matrix R_x is symmetric nonnegative definite, then it can be factorized as

$$R_x=R_x^{1/2}\left(R_x^{1/2}\right)^T. \tag{4}$$

Substituting (4) into (3) gives rise to

$$\begin{aligned} & E\left\{\|s - Tx\|^2\right\} \\ & = \operatorname{tr}\left\{(T - T_0) R_x^{1/2}\left(R_x^{1/2}\right)^T (T - T_0)^T\right\} \\ & \quad + E\left\{\|s - T_0 x\|^2\right\} \\ & = \operatorname{tr}\left\{\left[(T - T_0) R_x^{1/2}\right]\left[(T - T_0) R_x^{1/2}\right]^T\right\} \\ & \quad + E\left\{\|s - T_0 x\|^2\right\} \\ & = \left\|(T - T_0) R_x^{1/2}\right\|_F^2 + E\left\{\|s - T_0 x\|^2\right\} \\ & = \left\|TR_x^{1/2} - T_0 R_x^{1/2}\right\|_F^2 + E\left\{\|s - T_0 x\|^2\right\} \\ & = \left\|T_0 R_x^{1/2} - TR_x^{1/2}\right\|_F^2 + E\left\{\|s - T_0 x\|^2\right\}, \end{aligned} \tag{5}$$

since $E\{\| s - T_0 x\|^2\}$ is a constant, then

$$\begin{aligned} & \min_{T \in R^{m \times n},\ \operatorname{rank}(T) = d} E\left\{\|s - Tx\|^2\right\} \\ & = \min_{T \in R^{m \times n},\ \operatorname{rank}(T) = d} \left\|T_0 R_x^{1/2} - TR_x^{1/2}\right\|_F^2 + E\left\{\|s - T_0 x\|^2\right\}, \end{aligned} \tag{6}$$

that is to say, minimizing $E\{\|s - Tx\|^2\}$ is equivalent to minimizing $\|T_0 R_x^{1/2} - TR_x^{1/2}\|_F^2$. Therefore, we can find the solution T^* of (1) by solving the minimization problem

$$\min_{T \in R^{m \times n},\ \operatorname{rank}(T) = d} \left\|T_0 R_x^{1/2} - TR_x^{1/2}\right\|_F, \tag{7}$$

which can be summarized as the following low rank approximation problem:

Problem 1. Given two matrices $A \in R^{m \times n}$, $B \in R^{p \times n}$ and an integer d, $1 \leq d < m, p$, find a matrix $\widehat{X} \in R^{m \times p}$ of rank d such that

$$\left\|A - \widehat{X}B\right\|_F = \min_{X \in R^{m \times p},\ \operatorname{rank}(X) = d} \|A - XB\|_F. \tag{8}$$

In the last few years there has been a constantly increasing interest in developing the theory and numerical approaches for the low rank approximations of a matrix, due to their wide applications. A well-known method for the low rank approximation is the singular value decomposition (SVD) [5, 6]. When the desired rank is relatively low and the matrix is large and sparse, a complete SVD becomes too expensive. Some less expensive alternatives for numerical computation, for example, Lanczos bidiagonalization process [7], and the Monte Carlo algorithm [8] are available. To speed up the computation of SVD, random sampling has been employed in [9]. Recently, Ye [10] proposed the generalized low rank approximations of matrices (GLRAM) method. This method is proved to have less computational time than the traditional singular value decomposition-based methods in practical applications. Later, GLRAM method has been revisited and extended by Liu et al. [11] and Liang and Shi [12]. In some applications, we need to emphasize important parts and deemphasize unimportant parts of the data matrix, so the weighted low rank approximations were considered by many authors. Some numerical methods, such as Newton-like algorithm [13], left versus right representations method [14], and unconstrained optimization method [15], are proposed. Recently, by using the hierarchical identification principle [16] which regards the known matrix as the system parameter matrix to be identified, Ding et al. and Xie et al. present the gradient-based iterative algorithms [16–21] and least-squares-based iterative algorithm [22, 23] for solving matrix equations. The methods are innovational and computationally efficient numerical algorithms.

The common and practical method to tackle the low rank approximation Problem 1 is the singular value decomposition (SVD) (e.g. [1]). We briefly review SVD method as following. Minimizing (8) by a rank-d matrix XB is known [5, Page 69] to satisfy

$$XB = A_d = \sum_{i=1}^{d} \sigma_i u_i v_i^T, \tag{9}$$

where A_d denotes rank-d singular value decomposition truncation, that is, if the following SVD holds

$$A = \sum_{i=1}^{\operatorname{rank}(A)} \sigma_i u_i v_i^T, \tag{10}$$

then $A_d = \sum_{i=1}^{d} \sigma_i u_i v_i^T$. If the matrix B is square and nonsingular, then by (9) we obtain that the solution of Problem 1 is

$$X = A_d B^{-1} = \left(\sum_{i=1}^{d} \sigma_i u_i v_i^T\right) B^{-1}. \tag{11}$$

The SVD method has two disadvantages as following: (1) it requires the matrix B to be square and nonsingular; (2) in order to derive the solution (11), we must compute the inverse matrix of B, whose computation cost is very expensive.

In this paper, we develop a new method to solve the low rank approximation Problem 1, which can avoid the disadvantages of SVD method. We first transform Problem 1 into the fixed rank solution of a matrix equation and then use the generalized singular value decomposition (GSVD) to solve it. Based on these, we derive a sufficient condition for the existence of a solution of Problem 1, and the analytical expression of the solution is given. A numerical algorithm is proposed to compute the solution. Numerical examples are used to illustrate the numerical algorithm. The first one is artificial to show that the new algorithm is feasible to solve Problem 1, and the second is simulation, which shows that the new algorithm can be used to realize the image compression.

2. Main Results

In this section, we give a sufficient condition and an analytical expression for the solution of Problem 1 by transforming

Problem 1 into the fixed rank solution of a matrix equation. Finally, we establish an algorithm for solving Problem 1.

Lemma 2. *A matrix* $\widehat{X} \in R^{m\times n}$ *is a solution of Problem 1 if and only if it is a solution of the following matrix equation:*

$$XBB^T = AB^T, \qquad \operatorname{rank}(X) = d. \tag{12}$$

Proof. It is easy to verify that a matrix $\widehat{X} \in R^{m\times n}$ is a solution of Problem 1 if and only if $\widehat{X}$ satisfies the following two equalities simultaneously:

$$\left\| A - \widehat{X}B \right\|_F = \min_{X\in R^{m\times p}} \|A - XB\|_F, \tag{13}$$

$$\operatorname{rank}(X) = d. \tag{14}$$

Since the normal equation of the least squares problem (13) is

$$XBB^T = AB^T \tag{15}$$

and noting that the least squares problem (13) and its normal equation (15) have the same solution sets, then (13) and (14) can be equivalently written as

$$XBB^T = AB^T, \qquad \operatorname{rank}(X) = d \tag{16}$$

which also imply that Problem 1 is equivalent to (12). □

Remark 3. From Lemma 2 it follows that Problem 1 is equivalent to (12), hence we can solve Problem 1 by finding a fixed rank solution of the matrix equation $XBB^T = AB^T$.

Now we will use generalized singular value decomposition (GSVD) to solve (12). Set

$$C = BB^T \in R^{p\times p}, \qquad D = AB^T \in R^{m\times p}. \tag{17}$$

The GSVD of the matrix pair (C, D) is given by (see [24])

$$C = U\Sigma_1 W, \qquad D = V\Sigma_2 W, \tag{18}$$

where $U \in O^{p\times p}$, $V \in O^{m\times m}$, $W \in R^{p\times p}$ is a nonsingular matrix, $k = \operatorname{rank}([C^T, D^T])$, $r = \operatorname{rank}(C)$, $t = \operatorname{rank}(C) + \operatorname{rank}(D) - \operatorname{rank}([C^T, D^T])$, and

$$\begin{gathered}
\Sigma_1 = \begin{array}{c c}
\left(\begin{array}{cccc} I_C & 0 & 0 & 0 \\ 0 & S_C & 0 & 0 \\ 0 & 0 & O_C & 0 \end{array}\right) & \begin{array}{c} r-t \\ t \\ p-r, \end{array} \\
\begin{array}{cccc} r-t & t & k-r & p-k \end{array} &
\end{array} \\
\Sigma_2 = \begin{array}{c c}
\left(\begin{array}{cccc} O_D & 0 & 0 & 0 \\ 0 & S_D & 0 & 0 \\ 0 & 0 & I_D & 0 \end{array}\right) & \begin{array}{c} m-k-t+r \\ t \\ k-r \end{array} \\
\begin{array}{cccc} r-t & t & k-r & p-k \end{array} &
\end{array}
\end{gathered} \tag{19}$$

are block matrices, with I_C and I_D are identity matrices, O_C and O_D are zero matrices:

$$\begin{gathered}
S_C = \operatorname{diag}(\alpha_1, \alpha_2, \ldots, \alpha_t), \quad 1 > \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_t > 0, \\
S_D = \operatorname{diag}(\beta_1, \beta_2, \ldots, \beta_t), \quad 0 < \beta_1 \leq \beta_2 \leq \cdots \leq \beta_t < 1, \\
\alpha_i^2 + \beta_i^2 = 1, \quad i = 1, 2, \ldots, t.
\end{gathered} \tag{20}$$

By (17) and (18), we have

$$\begin{aligned}
XBB^T = AB^T &\Longleftrightarrow XC = D \Longleftrightarrow XU\Sigma_1 W = V\Sigma_2 W \\
&\Longleftrightarrow XU\Sigma_1 W - V\Sigma_2 W = 0 \\
&\Longleftrightarrow V\left(V^T XU\Sigma_1 - \Sigma_2\right) W = 0.
\end{aligned} \tag{21}$$

Set

$$Y = V^T XU, \tag{22}$$

and Y is partitioned as follows:

$$Y = \begin{array}{c c}
\left(\begin{array}{ccc} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{array}\right) & \begin{array}{c} m-k-t+r \\ t \\ k-r, \end{array} \\
\begin{array}{ccc} r-t & t & p-r \end{array} &
\end{array} \tag{23}$$

then

$$\begin{aligned}
&V^T XU\Sigma_1 - \Sigma_2 \\
&\quad = Y\Sigma_1 - \Sigma_2 \\
&\quad = \begin{pmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{pmatrix} \begin{pmatrix} I_C & 0 & 0 & 0 \\ 0 & S_C & 0 & 0 \\ 0 & 0 & O_C & 0 \end{pmatrix} \\
&\qquad - \begin{pmatrix} O_D & 0 & 0 & 0 \\ 0 & S_D & 0 & 0 \\ 0 & 0 & I_D & 0 \end{pmatrix} \\
&\quad = \begin{array}{c c}
\left(\begin{array}{cccc} Y_{11} & Y_{12}S_C & 0 & 0 \\ Y_{21} & Y_{22}S_C - S_D & 0 & 0 \\ Y_{31} & Y_{32}S_C & -I_D & 0 \end{array}\right) & \begin{array}{c} m-k-t+r \\ t \\ k-r. \end{array} \\
\begin{array}{cccc} r-t & t & k-r & p-k \end{array} &
\end{array}
\end{aligned} \tag{24}$$

Therefore, by (21) and (24), we have

$$\begin{aligned}
XBB^T = AB^T &\Longleftrightarrow V \begin{pmatrix} Y_{11} & Y_{12}S_C & 0 & 0 \\ Y_{21} & Y_{22}S_C - S_D & 0 & 0 \\ Y_{31} & Y_{32}S_C & -I_D & 0 \end{pmatrix} W = 0 \\
&\Longleftrightarrow \operatorname{rank}\left[V \begin{pmatrix} Y_{11} & Y_{12}S_C & 0 & 0 \\ Y_{21} & Y_{22}S_C - S_D & 0 & 0 \\ Y_{31} & Y_{32}S_C & -I_D & 0 \end{pmatrix} W \right] = 0 \\
&\Longleftrightarrow \operatorname{rank}\left[\begin{pmatrix} Y_{11} & Y_{12}S_C & 0 & 0 \\ Y_{21} & Y_{22}S_C - S_D & 0 & 0 \\ Y_{31} & Y_{32}S_C & -I_D & 0 \end{pmatrix} \right] = 0 \\
&\Longleftrightarrow k - r = 0, \quad Y_{11} = Y_{21} = Y_{31} = Y_{12} \\
&\qquad = Y_{32} = 0, \qquad Y_{22} = S_D S_C^{-1},
\end{aligned} \tag{25}$$

that is to say, the matrix equation $XBB^T = AB^T$ has a solution if and only if

$$\begin{aligned}
k - r &= \operatorname{rank}\left(\left[C^T, D^T\right]\right) - \operatorname{rank}(C) \\
&= \operatorname{rank}\left(\left[BB^T, BA^T\right]\right) - \operatorname{rank}\left(BB^T\right) = 0,
\end{aligned} \tag{26}$$

and according to (22), we know that the expression of the solution is

$$X = VYU^T, \tag{27}$$

where

$$Y = \begin{array}{ccc} \left(\begin{array}{ccc} 0 & 0 & Y_{13} \\ 0 & S_D S_C^{-1} & Y_{23} \end{array}\right) & \begin{array}{c} m-t \\ t, \end{array} \\ \begin{array}{ccc} r-t & t & p-r \end{array} & \end{array} \tag{28}$$

$$Y_{13} \in R^{(m-t)\times(p-r)}, \qquad Y_{23} \in R^{t\times(p-r)}.$$

By (26)–(28) and noting that Y_{13} and Y_{23} are arbitrary matrices, we have

$$\begin{aligned} \min_{X\in R^{m\times p}} \operatorname{rank}(X) &= \min_{X\in R^{m\times p}} \operatorname{rank}\left(VYU^T\right) \\ &= \min_{Y_{13}\in R^{(m-t)\times(p-r)}, Y_{23}\in R^{t\times(p-r)}} \operatorname{rank}(Y) \\ &= t \\ &= \operatorname{rank}(D) \\ &= \operatorname{rank}\left(AB^T\right), \\ \max_{X\in R^{m\times p}} \operatorname{rank}(X) &= \max_{Y\in R^{m\times p}} \operatorname{rank}\left(VYU^T\right) \\ &= \max_{Y_{13}\in R^{(m-t)\times(p-r)},\ Y_{23}\in R^{t\times(p-r)}} \operatorname{rank}(Y) \\ &= t + \min\{p-r, m-t\} \\ &= \min\{m, p+t-r\} \\ &= \min\{m, p + \operatorname{rank}(C) + \operatorname{rank}(D) \\ &\qquad - \operatorname{rank}\left(\left[C^T, D^T\right]\right) - \operatorname{rank}(C)\} \\ &= \min\left\{m, p + \operatorname{rank}(D) - \operatorname{rank}\left(\left[C^T, D^T\right]\right)\right\} \\ &= \min\{m, p + \operatorname{rank}(D) - \operatorname{rank}(C)\} \\ &= \min\left\{m, p + \operatorname{rank}\left(AB^T\right) - \operatorname{rank}\left(BB^T\right)\right\}. \end{aligned} \tag{29}$$

Hence, if

$$\operatorname{rank}\left(\left[BB^T, BA^T\right]\right) - \operatorname{rank}\left(BB^T\right) = k - r = 0, \tag{30}$$

$$\operatorname{rank}\left(AB^T\right) \le d \le \min\left\{m, p + \operatorname{rank}\left(AB^T\right) - \operatorname{rank}\left(BB^T\right)\right\}, \tag{31}$$

then (12) has a solution, and the expressions of the solution are given by (26)–(28), that is,

$$X = VYU^T = V\begin{pmatrix} 0 & 0 & Y_{13} \\ 0 & S_D S_C^{-1} & Y_{23} \end{pmatrix} U^T, \tag{32}$$

where $Y_{23} \in R^{t\times(p-r)}$ is an arbitrary matrix and $Y_{13} \in R^{(m-t)\times(p-r)}$ is chosen such that

$$\operatorname{rank}\left(Y_{13}\right) = d - t = d - \operatorname{rank}(D) = d - \operatorname{rank}\left(AB^T\right). \tag{33}$$

And noting that the low rank approximation Problem 1 is equivalent to (12) (i.e. Lemma 2), then we obtain the following.

Theorem 4. *If*

$$\begin{gathered} \operatorname{rank}\left(\left[BB^T, BA^T\right]\right) - \operatorname{rank}\left(BB^T\right) = k - r = 0, \\ \operatorname{rank}\left(AB^T\right) \le d \le \min\left\{m, p + \operatorname{rank}\left(AB^T\right) - \operatorname{rank}\left(BB^T\right)\right\}, \end{gathered} \tag{34}$$

then Problem 1 has a solution, and the expressions of the solution are given by

$$X = VYU^T = V\begin{pmatrix} 0 & 0 & Y_{13} \\ 0 & S_D S_C^{-1} & Y_{23} \end{pmatrix} U^T, \tag{35}$$

where $Y_{23} \in R^{t\times(p-r)}$ is an arbitrary matrix and $Y_{13} \in R^{(m-t)\times(p-r)}$ is chosen such that

$$\operatorname{rank}\left(Y_{13}\right) = d - t = d - \operatorname{rank}(D) = d - \operatorname{rank}\left(AB^T\right). \tag{36}$$

Remark 5. In contrast with (11), the solution expression (35) does not require the matrix B to be square and nonsingular and does not need to compute the inverse of B.

Based on Theorem 4, we can establish an algorithm for finding the solution of Problem 1.

Algorithm 6. (1) Input the matrices A, B and the integer d;

(2) make the GSVD of the matrix pair (C, D) according to (18);

(3) choose $Y_{23} \in R^{t\times(p-r)}$ and $Y_{13} \in R^{(m-t)\times(p-r)}$, such that $\operatorname{rank}(Y_{13}) = d - \operatorname{rank}(AB^T)$;

(4) compute the solution X according to (35).

3. Numerical Experiments

In this section, we first use a simple artificial example to illustrate that Algorithm 6 is feasible to solve Problem 1, then we use a simulation to show that Algorithm 6 can be used to realize the image compression. The experiments were done with MATLAB 7.6 on a 64-bit Intel Pentium Xeon 2.66 GHz with $e_{\text{mach}} \approx 2.0 \times 10^{-16}$.

(a) (b)

Figure 1: (a) Original image; (b) noisy image.

Table 1: Execution time for deriving Figures 2(a)–3(c).

Figure 2(a)	Figure 3(a)	Figure 2(b)	Figure 3(b)	Figure 2(c)	Figure 3(c)
3.5835 (s)	3.9216 (s)	2.8627 (s)	3.0721 (s)	2.0591 (s)	2.1433 (s)

Example 7. Consider Problem 1 with

$$\begin{aligned} A &= \begin{pmatrix} 0.6884 & 0.5873 & 0.4236 & 0.4243 & 0.8483 & 1.0400 & 0.9778 & 0.1541 \\ 1.0869 & 0.8243 & 0.5998 & 0.5993 & 1.0695 & 1.3080 & 1.5460 & 0.3538 \\ 0.5709 & 0.3925 & 0.1344 & 0.3478 & 0.4318 & 0.5389 & 0.4866 & 0.1035 \\ 1.1466 & 0.8938 & 0.9278 & 0.5209 & 1.3410 & 1.5086 & 1.4849 & 0.2599 \\ 0.7752 & 0.5851 & 0.4972 & 0.3909 & 0.8055 & 0.9389 & 0.9832 & 0.1978 \end{pmatrix}, \\ B &= \begin{pmatrix} 0.0554 & 0.6308 & 0.8468 & 0.9419 & 0.6891 & 0.2808 & 0.4558 & 0.7709 \\ 0.1655 & 0.4783 & 0.3702 & 0.7526 & 1.0548 & 0.4358 & 0.7835 & 0.3309 \\ 0.1150 & 0.3808 & 0.3316 & 0.5940 & 0.7677 & 0.3168 & 0.5645 & 0.2978 \\ 0.1693 & 0.5032 & 0.4001 & 0.7903 & 1.0890 & 0.4499 & 0.8073 & 0.3580 \\ 0.0869 & 0.4247 & 0.4608 & 0.6495 & 0.6779 & 0.2786 & 0.4829 & 0.4170 \\ 0.1142 & 0.5772 & 0.6351 & 0.8815 & 0.9040 & 0.3715 & 0.6421 & 0.5951 \end{pmatrix}. \end{aligned} \tag{37}$$

We make GSVD of the matrix pair $(C, D) = (BB^T, AB^T)$ as follows:

$$C = U\Sigma_1 W, \qquad D = V\Sigma_2 W, \tag{38}$$

where

$$U = \begin{pmatrix} -0.7574 & 0.4215 & -0.0590 & 0.0130 & -0.2832 & -0.4060 \\ -0.1509 & -0.5893 & -0.3965 & -0.6437 & -0.1562 & -0.1845 \\ -0.1775 & -0.3461 & 0.8997 & -0.1454 & -0.0810 & -0.1071 \\ -0.1752 & -0.5847 & -0.1667 & 0.7506 & -0.1161 & -0.1510 \\ -0.3397 & -0.0817 & -0.0302 & -0.0251 & 0.9311 & -0.0970 \\ -0.4754 & -0.0814 & -0.0331 & -0.0218 & -0.0915 & 0.8703 \end{pmatrix}, \tag{39}$$

$$V = \begin{pmatrix} -0.2818 & 0.4322 & 0.0937 & 0.7418 & -0.4180 \\ -0.1814 & -0.6971 & -0.4417 & 0.0795 & -0.5503 \\ 0.9419 & -0.0182 & -0.0330 & 0.2445 & -0.2272 \\ 0.0166 & 0.5179 & -0.2279 & -0.5906 & -0.5751 \\ -0.0160 & -0.2424 & 0.8753 & -0.1863 & -0.3743 \end{pmatrix}, \tag{40}$$

$$W = \begin{pmatrix} -6.2034 & -5.7059 & -4.4070 & -5.9640 & -4.5789 & -6.1909 \\ -7.5279 & -8.6034 & -6.4574 & -8.9389 & -6.2317 & -8.3756 \\ 0.4175 & -0.1314 & 0.2453 & 0.4166 & -0.1876 & -0.7345 \\ 0.1928 & 0.6688 & 0.2243 & -0.6061 & -0.2199 & -0.2228 \\ -0.0101 & 0.2451 & -0.4669 & -0.0248 & 0.7439 & -0.4097 \\ 0.1926 & 0.3245 & -0.7213 & 0.2930 & -0.4910 & 0.1022 \end{pmatrix}, \tag{41}$$

$$\Sigma_1 = \begin{pmatrix} 0.9985 & 0 & 0 & 0 & 0 \\ 0 & 0.4192 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.0541 & 0 & 0 & 0 & 0 \\ 0 & 0.9079 & 0 & 0 & 0 \end{pmatrix}. \tag{42}$$

It is easy to verify that

$$\begin{aligned} &\operatorname{rank}\left(\left[BB^T, BA^T\right]\right) - \operatorname{rank}\left(BB^T\right) = 0, \\ &\operatorname{rank}\left(AB^T\right) = 2, \\ &\min\left\{m, p + \operatorname{rank}\left(AB^T\right) - \operatorname{rank}\left(BB^T\right)\right\} = 5, \end{aligned} \tag{43}$$

that is, if $2 \le d \le 5$, then the conditions of Theorem 4 are satisfied. Setting $d = 2 \in [2, 5]$, according to (35), we obtain that the solution of Problem 1 is

$$\widehat{X} = V \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dfrac{0.0541}{0.9985} & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{0.9079}{0.4192} & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{44}$$

$$U^T = \begin{pmatrix} -0.4121 & 0.5275 & 0.3062 & 0.5223 & 0.0603 & 0.0546 \\ -0.5057 & 0.7017 & 0.4117 & 0.6961 & 0.0959 & 0.0949 \\ -0.2175 & 0.2880 & 0.1680 & 0.2855 & 0.0357 & 0.0338 \\ -0.5008 & 0.7389 & 0.4367 & 0.7339 & 0.1126 & 0.1166 \\ -0.3341 & 0.4793 & 0.2823 & 0.4758 & 0.0696 & 0.0708 \end{pmatrix}. \tag{45}$$

Setting $d = 4 \in [2, 5]$, according to (35), we obtain that the solution of Problem 1 is

$$\widehat{X} = V \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dfrac{0.0541}{0.9985} & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{0.9079}{0.4192} & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$U^T = \begin{pmatrix} -0.3898 & 0.3610 & -0.0102 & 0.8937 & 0.0579 & 0.0549 \\ -0.5040 & 1.2224 & 0.3498 & 0.2032 & 0.1188 & 0.1155 \\ -0.2733 & -0.0736 & 1.0181 & 0.1148 & 0.0077 & 0.0030 \\ -0.4950 & 0.3989 & 0.3764 & 1.1198 & 0.0991 & 0.1052 \\ -0.3363 & 0.6416 & 0.3032 & 0.2965 & 0.0762 & 0.0764 \end{pmatrix}. \tag{46}$$

Example 7 shows that Algorithm 6 is feasible to solve Problem 1. However, the SVD method in [1] cannot be used to solve Example 7, because B is not a square matrix.

Example 8. We will use the generalized Karhunen-Loeve transform, based on Algorithm 6 and SVD method in [1], respectively, to realize the image compression. Figure 1(a) (see page 3) is the test image which has 256 × 256 pixels and 256 levels on each pixel. We separate it into 32 × 32 blocks such that each block has 8 × 8 pixels. Let $f_{i,j}^{(k,l)}$ and $n_{i,j}^{(k,l)}$ ($i, j = 0, 1, 2, \ldots, 7$; $k, l = 0, 1, 2, \ldots, 31$) be the values of the image and a Gaussian noise (generated by Matlab function *imnoise*) at the (i, j)th pixel in the (k, l)th block, respectively. For convenience, let $a = i + 8j$, $p = k + 32l$, and the (i, j)th pixel in the (k, l)th block be expressed as the ath pixel in the pth block ($a = 0, 1, 2, \ldots, 63$; $p = 0, 1, \ldots, 1023$). We can also express $f_{i,j}^{(k,l)}$ and $n_{i,j}^{(k,l)}$ as $f_a^{(p)}$ and $n_a^{(p)}$, respectively.

The test image is processed on each block. Therefore, we can assume that the blocked image space is 64-D real vector space R^{64}. The pth block of the original image is expressed by the pth vector:

$$s^p = \left(s_0^p, s_1^p, \ldots, s_{63}^p\right)^T. \tag{47}$$

Hence the original image is expressed by 1024 64-D vectors $\{s^p\}_{p=0}^{1023}$. The noise is similarly expressed by $\{n^p\}_{p=0}^{1023}$, where

$$n^p = \left(n_0^p, n_1^p, \ldots, n_{63}^p\right)^T. \tag{48}$$

Figure 1(b) is the noisy image $\{x^p\}_{p=0}^{1023}$, where

$$x^p = s^p + n^p, \quad p = 0, 1, \ldots, 1023. \tag{49}$$

By (47), (49), (2), (4) and the definition of covariance matrix, we get T_0 and $R_x^{1/2}$ of (7). Then we use Algorithm 6 and SVD method in [1] to realize the image compression respectively, and the experiment results are in pages 4 and 5.

Figure 2 illustrates that Algorithm 6 can be used to realize image compression. Although it is difficult to see the difference between Figures 2 and 3, which are compressed by SVD method in [1], from Table 1 we can see that the execution time of Algorithm 6 is less than that of SVD method at the same rank. This shows that our algorithm outperforms the SVD method in execution time.

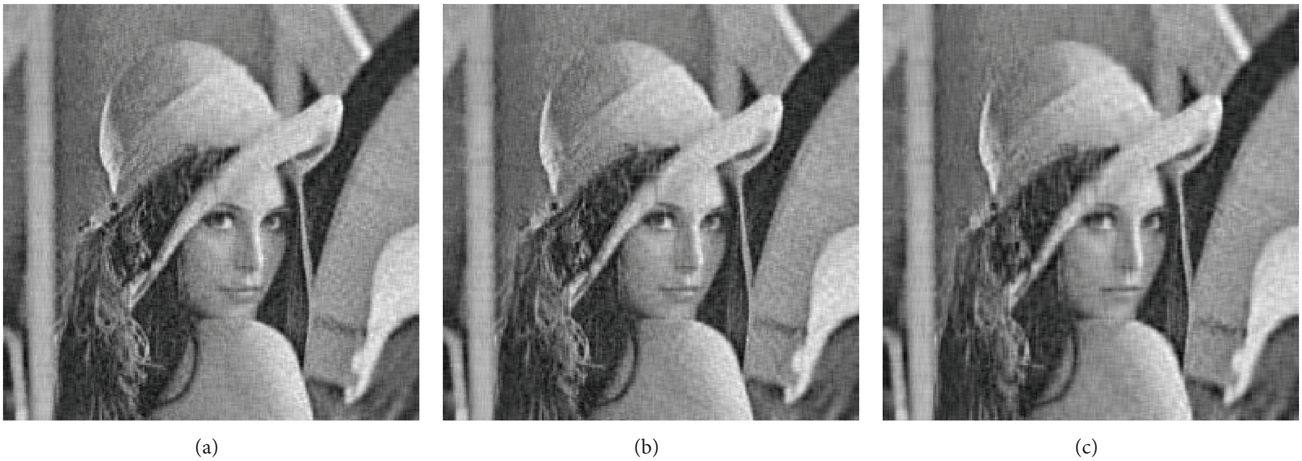


Figure 2: Image compression by Algorithm 6 with different rank d: (a) $d = 40$; (b) $d = 30$; and (c) $d = 20$.

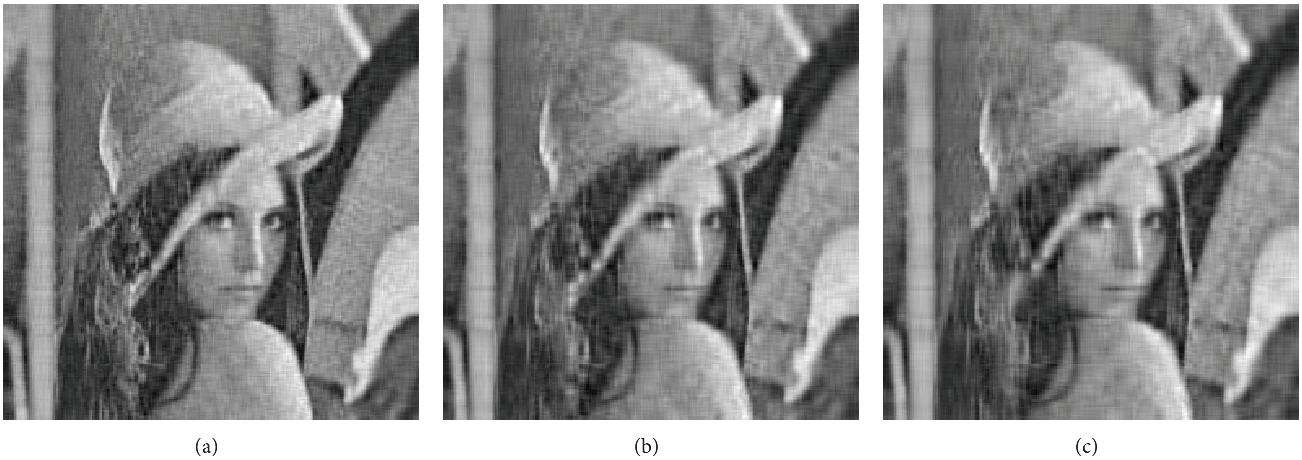


Figure 3: Image compression by SVD method with different rank d: (a) $d = 40$; (b) $d = 30$; and (c) $d = 20$.

4. Conclusion

The low rank approximation Problem 1 arising in the generalized Karhunen-Loeve transform is studied in this paper. We first transform Problem 1 into the fixed rank solution of a matrix equation and then use the generalized singular value decomposition (GSVD) to solve it. Based on these, we derive a sufficient condition for the existence of a solution, and the analytical expression of the solution is also given. Finally, we use numerical experiments to show that new algorithm is feasible and effective.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (11101100; 11226323; 11261014; and 11171205), the Natural Science Foundation of Guangxi Province (2012GXNSFBA053006; 2013GXNSFBA019009; and 2011GXNSFA018138), the Key Project of Scientific Research Innovation Foundation of Shanghai Municipal Education Commission (13ZZ080), the Natural Science Foundation of Shanghai (11ZR1412500), the Ph.D. Programs Foundation of Ministry of Education of China (20093108110001), the Discipline Project at the corresponding level of Shanghai (A. 13-0101-12-005), and Shanghai Leading Academic Discipline Project (J50101).

References

[1] Y. Hua and W. Q. Liu, “Generalized Karhunen-Loeve transform,” *IEEE Signal Processing Letters*, vol. 5, pp. 141–142, 1998.

[2] S. Kraut, R. H. Anderson, and J. L. Krolik, “A generalized Karhunen-Loeve basis for efficient estimation of tropospheric refractivity using radar clutter,” *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 48–60, 2004.

[3] H. Ogawa and E. Oja, “Projection filter, Wiener filter, and Karhunen-Loève subspaces in digital image restoration,” *Journal of Mathematical Analysis and Applications*, vol. 114, no. 1, pp. 37–51, 1986.

[4] Y. Yamashita and H. Ogawa, “Relative Karhumen-Loeve transform,” *IEEE Transactions on Signal Process*, vol. 44, pp. 371–378, 1996.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[6] P. C. Hansen, "The truncated SVD as a method for regularization," *BIT Numerical Mathematics*, vol. 27, no. 4, pp. 534–553, 1987.

[7] H. D. Simon and H. Zha, "Low-rank matrix approximation using the Lanczos bidiagonalization process with applications," *SIAM Journal on Scientific Computing*, vol. 21, no. 6, pp. 2257–2274, 2000.

[8] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo algorithms for matrices—II. Computing a low-rank approximation to a matrix," *SIAM Journal on Computing*, vol. 36, no. 1, pp. 158–183, 2006.

[9] A. Frieze, R. Kannan, and S. Vempala, "Fast Monte-Carlo algorithms for finding low-rank approximations," *Journal of the ACM*, vol. 51, no. 6, pp. 1025–1041, 2004.

[10] J. P. Ye, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, pp. 167–191, 2005.

[11] J. Liu, S. C. Chen, Z. H. Zhou, and X. Y. Tan, "Generalized low rank approximations of matrices revisited," *IEEE Transactions on Neural Networks*, vol. 21, pp. 621–632, 2010.

[12] Z. Z. Liang and P. F. Shi, "An analytical algorithm for generalized low rank approxiamtions of matrices," *Pattern Recognition*, vol. 38, pp. 2213–2216, 2005.

[13] J. H. Manton, R. Mahony, and Y. Hua, "The geometry of weighted low-rank approximations," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 500–514, 2003.

[14] I. Markovsky and S. Van Huffel, "Left versus right representations for solving weighted low-rank approximation problems," *Linear Algebra and its Applications*, vol. 422, no. 2-3, pp. 540–552, 2007.

[15] M. Schuermans, P. Lemmerling, and S. Van Huffel, "Block-row Hankel weighted low rank approximation," *Numerical Linear Algebra with Applications*, vol. 13, no. 4, pp. 293–302, 2006.

[16] F. Ding and T. Chen, "On iterative solutions of general coupled matrix equations," *SIAM Journal on Control and Optimization*, vol. 44, no. 6, pp. 2269–2284, 2006.

[17] F. Ding and T. Chen, "Gradient based iterative algorithms for solving a class of matrix equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.

[18] F. Ding, P. X. Liu, and J. Ding, "Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle," *Applied Mathematics and Computation*, vol. 197, no. 1, pp. 41–50, 2008.

[19] J. Ding, Y. Liu, and F. Ding, "Iterative solutions to matrix equations of the form AiXBi = Fi," *Computers & Mathematics with Applications*, vol. 59, no. 11, pp. 3500–3507, 2010.

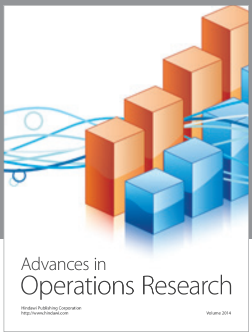
[20] L. Xie, Y. Liu, and H. Yang, "Gradient based and least squares based iterative algorithms for matrix equations $AXB + CX^TD = F$," *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 2191–2199, 2010.

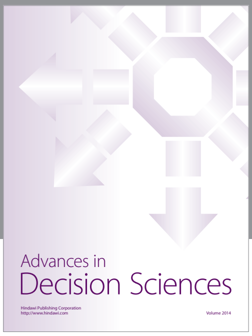
[21] L. Xie, J. Ding, and F. Ding, "Gradient based iterative solutions for general linear matrix equations," *Computers & Mathematics with Applications*, vol. 58, no. 7, pp. 1441–1448, 2009.

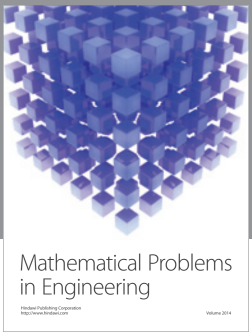
[22] F. Ding and T. Chen, "Iterative least-squares solutions of coupled Sylvester matrix equations," *Systems & Control Letters*, vol. 54, no. 2, pp. 95–107, 2005.

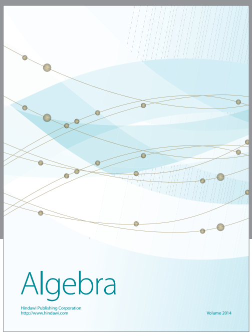
[23] W. Xiong, W. Fan, and R. Ding, "Least-squares parameter estimation algorithm for a class of input nonlinear systems," *Journal of Applied Mathematics*, vol. 2012, Article ID 684074, 14 pages, 2012.

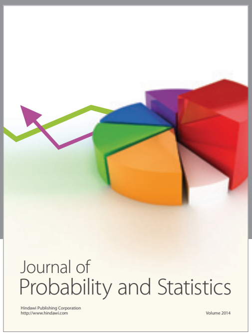
[24] C. C. Paige and M. A. Saunders, "Towards a generalized singular value decomposition," *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 398–405, 1981.


Advances in
Operations Research

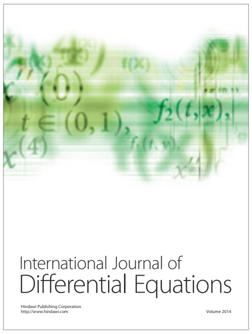

Advances in
Decision Sciences

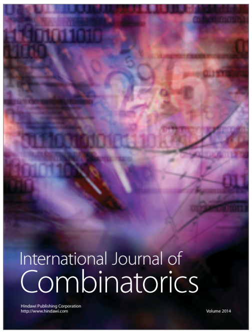

Mathematical Problems
in Engineering

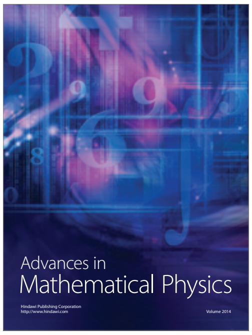

Algebra

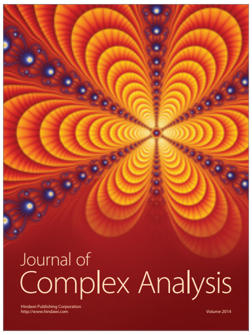

Journal of
Probability and Statistics

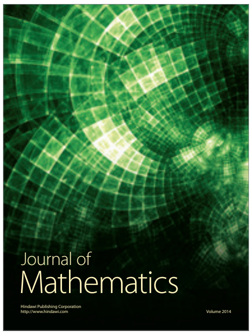

The Scientific
World Journal

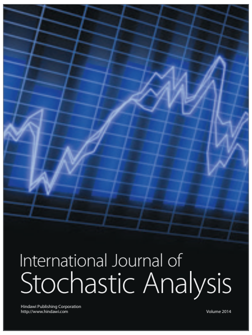

International Journal of
Differential Equations

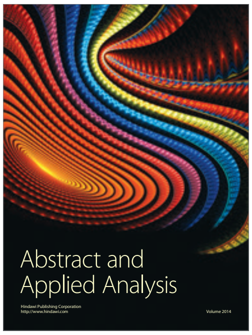

International Journal of
Combinatorics

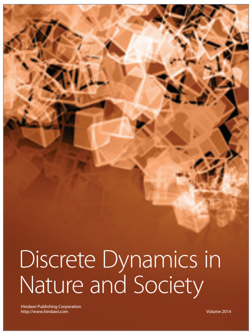

Advances in
Mathematical Physics

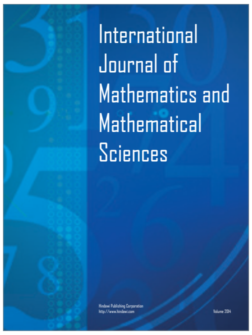

Journal of
Complex Analysis


Journal of
Mathematics


International Journal of
Stochastic Analysis

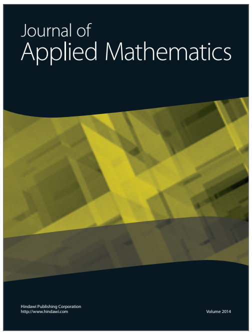

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences


Journal of
Discrete Mathematics


Journal of
Function Spaces


Journal of
Applied Mathematics


Journal of
Optimization