2-2010

A Majorization Algorithm for Constrained Correlation Matrix Approximation

Daniel J. Simon
General Motors Company, d.j.simon@csuohio.edu

Jeff Abell
University of California - Davis

A majorization algorithm for constrained correlation matrix approximation

Dan Simon [a,*], Jeff Abell [b]

[a] Cleveland State University, Department of Electrical and Computer Engineering, 2121 Euclid Avenue, Cleveland, Ohio 44115, United States
[b] General Motors Company, Global Research & Development, 30500 Mound Road, Warren, Michigan 48090, United States

1. Introduction

The approximation of an input matrix by a correlation matrix is a fundamental problem in applied mathematics. A correlation matrix is a symmetric positive semidefinite matrix with unit diagonal, and any symmetric positive semidefinite matrix with unit diagonal is a correlation matrix. Sometimes it is also desired that the correlation matrix be rank-deficient. Applications of this problem occur in finance [11], resource allocation [9], industrial process monitoring [20], image processing [18], reduced order state estimation [29], and quality function deployment [10]. Correlation matrix approximation is a special type of matrix nearness problem, of which a classic survey is given in [16].

The input matrix R is nominally a correlation matrix, but for a variety of reasons it might not be positive semidefinite [28]. First, the data used to generate the matrix might be incomplete, or might contain noise and outliers that pollute the matrix. Second, the data used to generate the matrix might be asynchronous. Third, the matrix might be adjusted by humans on the basis of intuition. All of these factors, and possibly others, could give rise to a matrix that is not positive semidefinite, but that humans intend to use as a correlation matrix. This gives rise to the problem of finding a correlation matrix that is as close as possible to the given indefinite matrix.

Recall that any $n \times n$ correlation matrix with rank d can be decomposed as $\widehat{R} = XX^T$, where X is an $n \times d$ matrix. This is called the Cholesky decomposition of $\widehat{R}$ [4]. We want to find the matrix $\widehat{R}$ that is as close to possible to some $n \times n$ symmetric input matrix R, which might not be positive semidefinite. We assume that both R and $\widehat{R}$ have ones on the diagonal.

This problem has been solved in several ways. First, perhaps the simplest way is principal component analysis (PCA) [13]. In this approach we find the Jordan form decomposition $R = Q \Lambda Q^T$, where the columns of the orthogonal matrix Q contain the eigenvectors of R, and Λ is a diagonal matrix that contains the eigenvalues of R. Assuming that the eigenvalues are in descending order, we define Q_d as the first d columns of Q, and Λ_d as the upper-left $d \times d$ block of Λ, where d is the number of positive eigenvalues of R. Denote the n rows of $Q_d \Lambda_d^{1/2}$ as $z_1, \ldots, z_n$. Find X as

$$X = \begin{bmatrix} z_1^T/\|z_1\|_2 & \cdots & z_n^T/\|z_n\|_2 \end{bmatrix}^T. \tag{1}$$

X is an $n \times d$ matrix, and XX^T is an $n \times n$ matrix with d positive eigenvalues and $(n - d)$ zero eigenvalues. PCA gives the least squares approximation to R, and it is relatively simple and straightforward, but it cannot be extended to weighted least squares or constrained least squares.

The second approach to solving the correlation matrix approximation problem is geometric optimization [15], which is based on Riemannian geometry and the mathematics of topology and manifolds. This approach solves the problem

$$\min_X \|R - XX^T\| \tag{2}$$

for any seminorm $\| \cdot \|$. The approach is actually even more general and can be used to minimize any sufficiently smooth objective function, although the algorithm is quite complicated. The objective function might have multiple local minima, but the geometric approach includes a way of checking if a local minimum is global. In [15] geometric optimization is compared with majorization [26], parameterization [28], alternating projections [14], Lagrange multipliers [30], and Matlab's "fmincon" function (a general constrained function minimizer). The results presented in [15] indicated that geometric optimization was the most efficient algorithm for the test cases considered.

The third approach applies only to problems in which the input matrix R contains correlations between financial currencies [8]. This assumes that R is already positive definite. If the data from which R was obtained are changed, a statistical method is given to modify R in order to ensure that it is consistent with the changed data, and the modified matrix remains positive definite. This method is not optimal in any sense.

The fourth approach, called hypersphere decomposition, is a trigonometric parameterization of the elements of X, recognizing that if XX^T is a correlation matrix, then the rows of X can be viewed as vectors in a unit hypersphere [28]. The correlation matrix approximation problem is thus transformed to a parametric optimization problem in which correlation matrix related constraints are naturally and automatically satisfied due to the parameterization. An extension of this approach which uses Jacobi rotations to reduce the parameterization and thus reduce computational effort is discussed in [7].

The fifth approach uses alternating projections based on convex analysis [17]. This approach is based on iteratively projecting R onto the closest positive semidefinite matrix, and then setting the diagonal elements of the result to ones. It is shown that (2) has a unique solution if the Frobenius norm is used, as long as $\widehat{R} = XX^T$ is full rank. Local linear convergence of the alternating projections method is guaranteed.

The sixth approach is to pose the approximation problem as a constrained semidefinite programming problem [2]. This problem can then be solved using a specially formulated interior-exterior point algorithm designed specifically for sparse R. Local quadratic convergence is guaranteed, although

computational cost is quite high. The search direction of this iterative algorithm is determined with a Gauss-Newton method.

The seventh approach [12] is based on the decomposition $X = \Lambda Z$ where Λ is diagonal and each row of Z has unit length. Then optimization is alternated over elements of Λ and rows of Z. This method is particularly suited if (2) is changed to a weighted Frobenius norm and the weighting matrix has certain special properties.

The eighth approach is a method to generate random numbers from a general symmetric pseudo-correlation matrix R (possibly indefinite) [1]. The method is based on a Cholesky factorization of R which is modified to work with indefinite R matrices. The experimental correlation of the Monte Carlo generation of random numbers implicitly generates a positive semidefinite approximation to R.

The ninth approach is to use a random walk method to solve the approximation problem [21]. This is very general and can be used with any norm in (2). It uses PCA to find an initial guess for $\widehat{R}$, and then uses a random walk on a particular factorization of $\widehat{R}$ to minimize the specified norm. The method is demonstrated in [21] with the least maximum norm and the Frobenius norm. However, random walk is an ad-hoc optimization approach that is relatively simple and slow.

The 10th approach is to use methods from computer intelligence to solve the approximation problem. For example, differential evolution can be used to either fill in an incomplete correlation matrix [22] or to find an approximating correlation matrix [23]. The idea that is proposed in [22,23] uses differential evolution, but it can be adapted to any population based optimization method, such as a genetic algorithm, particle swarm optimization, ant colony optimization, etc. This approach can take advantage of the results of computer intelligence, but population based optimizers are typically slow and require a lot of tuning. Correlation matrix approximation using differential evolution is demonstrated in [23] for the absolute norm, the Frobenius norm, and the Chebyshev norm.

The 11th approach is to combine gradient descent with Lagrange multiplier methods [30] to convert a constrained optimization problem into an unconstrained min–max problem to obtain a low rank approximation of R. The maximizing problem is solved with closed form spectral decomposition, and the minimization problem is solved with gradient descent.

The 12th approach is to project R onto the closest matrix with ones on the diagonal, and then project the result onto a semidefinite matrix of rank d or less [14,15,25]. This method does not necessarily converge to a minimum of the problem, but it always converges to a feasible solution.

The 13th approach is presented in [27] and relies on the theory of strongly semismooth matrix valued functions. It can minimize (2) subject to $\widehat{R}_{ij} = 0$ constraints by using a Lagrangian dual approach to extend a generalized Newton's method. In fact, the constraints do not have to be zeros, they can be any constants that are consistent with a feasible solution. This method is closest to solving the problem discussed in this paper. However it has not been generalized to weighted minimization of the type discussed below in (3), and it has not been generalized to low rank approximations of R. That is, $\widehat{R} = XX^T$ is assumed to be full rank. Low rank approximations to R may be particularly desirable for applications such as data compression and image processing.

Finally we discuss correlation matrix approximation using majorization [5, Section 8.4, 26]. In general, a majorization algorithm to minimize some function $f(x)$ can be summarized as follows.

1. Initialize x_0 as an initial guess for the minimizing value of x. Initialize the iteration number $k = 0$.
2. Find a function $g_k(x)$, called a majorization function, that satisfies three criteria.

   (a) First, $g_k(x)$ is analytically simpler than $f(x)$.
   (b) Second, $g_k(x_k) = f(x_k)$.
   (c) Third, $g_k(x) \geqslant f(x)$ for all x.

3. Minimize $g_k(x)$. The minimizing value of x is equal to x_{k+1}. Increment k by one and go to step 2 for the next iteration.

The iteration can be terminated after $g_k(x)$ converges within some tolerance, or if its derivative falls below some threshold, or if the iteration count exceeds some threshold. This algorithm guarantees that $f(x_{k+1}) \leqslant f(x_k)$. An example of a few iterations are illustrated in Fig. 1.
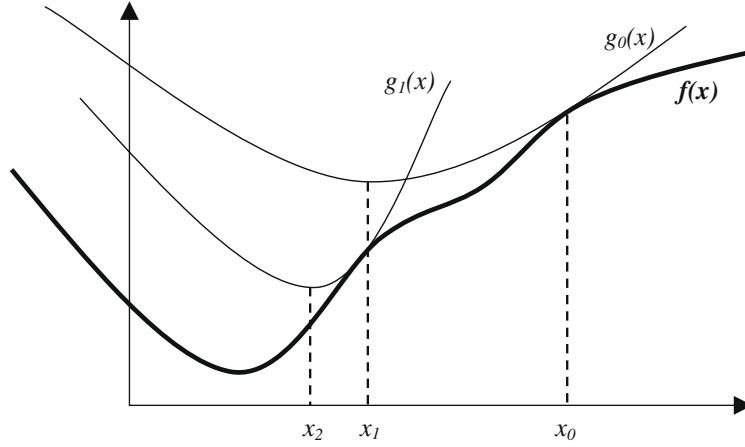
Fig. 1. Illustration of majorization, adapted from [26]. The minimization process starts with an initial guess x_0. $g_0(x)$ is a function that matches $f(x)$ at $x = x_0$. We minimize $g_0(x)$ to obtain x_1. $g_1(x)$ is a function that matches $f(x)$ at $x = x_1$. We minimize $g_1(x)$ to obtain x_2. The process continues until convergence.

An algorithm is presented in [26] to find an approximating correlation matrix using majorization. The algorithm has the flexibility to minimize a weighted Frobenius norm of the difference between the input and output matrices. The optimization problem can be written as

$$\min_X f(X), \quad \text{where}$$
$$f(X) = \parallel \widetilde{W} \circ (R - XX^T) \parallel_F^2 = \sum_i \sum_{j \neq i} W_{ij}(R_{ij} - X_i X_j^T)^2, \tag{3}$$

where $A \circ B$ is the Hadamard product of A and B, $\widetilde{W}$ is a symmetric weighting matrix, $W_{ij} = \widetilde{W}_{ij}^2$, and X_i is the ith row of X. We used the fact that $R_{ii} = X_i X_i^T = 1$ (by assumption) to derive the third part of (3).

Now suppose that we want to find a rank deficient correlation matrix $\widehat{R}$ to approximate R. In that case we want to solve (3) for the $n \times d$ matrix X. This will give an approximation XX^T to R such that XX^T has d nonnegative eigenvalues, plus an addition $n - d$ zero eigenvalues.

We might also want to solve (3) subject to the constraint that specified elements of XX^T are zero. In many cases an algorithm or human decision maker tries to generate a correlation matrix on the basis of some data, but the resulting matrix R is not quite positive semidefinite. An algorithm is needed to find a matrix X such that XX^T (which is positive semidefinite by construction) is close to R. However, it is desired that if $R_{ij} = 0$, then $(XX^T)_{ij} = 0$ also. This constraint could arise for one of several reasons. For example, the human decision maker may know a priori that the correlation between two random variables is zero, so it would be nonintuitive and aesthetically unpleasant to see nonzero values at certain elements of XX^T. Another reason for enforcing zeros at specific locations in XX^T may be to reduce subsequent computational effort associated with calculations that use the matrix. The problem of (3) can therefore be modified to

$$\min_X f(X) \quad \text{such that } X_i X_j^T = 0 \quad \text{for all } (i,j) \in S, \tag{4}$$

where S is a user-specified set of row/column indices. Although many correlation matrix approximation approaches have been proposed as summarized earlier in this paper, this specific correlation matrix approximation problem, with the low rank specification and the $\widehat{R}_{ij} = 0$ constraints, has not been studied until now.

2. Constrained correlation matrix approximation

In this section we propose a solution to (4) using majorization. The approach presented here closely parallels [26]. Note that $f(X)$ in (3) can be written as

$$
\begin{aligned}
f(X) &= \sum_i \sum_{j \neq i} W_{ij} \left(R_{ij}^2 + \left(X_i X_j^T \right)^2 - 2 R_{ij} X_i X_j^T \right) \\
&= c_1 + \sum_i \sum_{j \neq i} \left(W_{ij} \left(X_i X_j^T \right) \left(X_i X_j^T \right) - 2 W_{ij} R_{ij} X_i X_j^T \right) \\
&= c_1 + \sum_i \sum_{j \neq i} \left(W_{ij} \left(X_i X_j^T \right) \left(X_j X_i^T \right) - 2 W_{ij} R_{ij} X_i X_j^T \right) \\
&= c_1 + \sum_i \sum_{j \neq i} \left(W_{ij} X_i \left(X_j^T X_j \right) X_i^T - 2 W_{ij} R_{ij} X_i X_j^T \right),
\end{aligned}
\tag{5}
$$

where c_1 is constant with respect to X. The above can be written as

$$
\begin{aligned}
f(X) &= c_1 + \sum_i X_i \left(\sum_{j \neq i} W_{ij} X_j^T X_j \right) X_i^T - 2 \sum_i X_i \left(\sum_{j \neq i} W_{ij} R_{ij} X_j^T \right) \\
&= c_1 + \sum_i X_i B_i X_i^T - 2 \sum_i X_i \left(\sum_{j \neq i} W_{ij} R_{ij} X_j^T \right),
\end{aligned}
\tag{6}
$$

where B_i is defined by the above equation. Now treat $f(X)$ as a function of X_i, where all the rows of X besides X_i are fixed. We can rewrite the above equation as

$$
f(X) = c_1 + \sum_i f_i(X_i),
$$

$$
f_i(x) = x B_i x^T - 2x \left(\sum_{j \neq i} W_{ij} R_{ij} X_j^T \right).
\tag{7}
$$

Now we will find a majorization function for $f_i(x)$. Suppose that λ_i is the largest eigenvalue of B_i. Then $B_i - \lambda_i I$ is negative semidefinite and

$$
(x - X_i)(B_i - \lambda_i I)(x - X_i)^T \leq 0
\tag{8}
$$

for any $1 \times d$ vector x. If $\|x\|_2 = 1$ then the above can be written as

$$
x B_i x^T \leq 2\lambda_i - 2x(\lambda_i X_i^T - B_i X_i^T) - X_i B_i X_i^T,
\tag{9}
$$

with equality for $x = X_i$. Combining this inequality with (7) gives

$$
f_i(x) \leq g_i(x),
\tag{10}
$$

with equality for $x = X_i$, where $g_i(x)$ is given as

$$
g_i(x) = c_2 - 2x \left(\lambda_i X_i^T - B_i X_i^T + \sum_{j \neq i} W_{ij} R_{ij} X_j^T \right),
\tag{11}
$$

where c_2 is constant with respect to x. We see that $g_i(x)$ is a majorization function for $f_i(x)$. The function $g_i(x)$ is an attractive majorization function because it is linear in x and therefore easy to minimize. Our majorization approach to minimizing $f(X)$ is to minimize $g_i(x)$ for each i, and then repeat until convergence.

Now recall that S is the set of (i, j) indices in XX^T such that the constraint $X_i X_j^T = 0$ holds. To set up the constrained minimization of $g_i(x)$, we define S_i as the set of column indices in S that are less than their corresponding row indices. That is,

$$S_i = \{j : j < i, (i,j) \in S\}. \tag{12}$$

The constrained minimization of $g_i(x)$ can be written as

$$\min_x \left\{ g_i(x) : \| x \|_2 = 1, x X_j^T = 0 \text{ for all } j \in S_i \right\}. \tag{13}$$

The constraint $x X_j^T = 0$ for all $j \in S_i$ means that

$$x = \sum_k \alpha_k v_k, \tag{14}$$

where $\{\alpha_k\}$ is a set of unknown constants, and $\{v_k\}$ is a set of linearly independent row vectors that form a basis for the subspace that is orthogonal to X_j. According to the principle of orthogonality [24], the solution of (13) can be written as

$$z = \lambda_i X_i^T - B_i X_i^T + \sum_{j \neq i} W_{ij} R_{ij} X_j^T,$$

$$q = \sum_k \frac{v_k z}{v_k v_k^T} v_k,$$

$$x = q / \| q \|_2. \tag{15}$$

Implementing (15) requires finding the vectors $\{v_k\}$. This can be done several ways, among which is QR decomposition [24]. First we find the X_j vectors for which $j \in S_i$. Suppose that there are n_i of these vectors, each containing d elements. We collect all of these vectors in a $d \times n_i$ matrix A.

$$A = \begin{bmatrix} X_{j(1)}^T & \cdots & X_{j(n_i)}^T \end{bmatrix}, \tag{16}$$

where $j(k)$ is the kth element of S_i. Now suppose that the rank of A is equal to m (note that $m \leq n_i < d$). QR decomposition finds a $d \times d$ matrix Q and a $d \times n_i$ upper triangular matrix Υ such that $A = Q\Upsilon$. The first m columns of Q form an orthogonal basis for the columns of A, and the last $d - m$ columns of Q form an orthogonal basis for the subspace that is orthogonal to the columns of A. The transposes of the last $d - m$ columns of Q therefore comprise the v_k vectors of (15). The upper index of the k summations in (14) and (15) is therefore $d - m$.

Note that if S_i is empty, then this algorithm reduces to that given in [26]. That is, (15) becomes

$$x = z / \| z \|_2. \tag{17}$$

Based on these ideas, a majorization algorithm for constrained correlation matrix approximation can be given as follows.

Algorithm 1. The problem solved by this algorithm is to find an $n \times d$ matrix X such that $\widehat{R} = XX^T$ minimizes (3), subject to the constraints $\widehat{R}_{ii} = 1$ for all i, and $\widehat{R}_{ij} = 0$ for all $\{i,j\} \in S$. (Typically $S = \{i,j\} : R_{ij} = 0$.)

1. Use the PCA method of (1), or some other method, to find an initial guess for X, denoted $X^{(0)}$. Note that $X^{(0)}$ need not necessarily satisfy the constraint $\widehat{R}_{ij} = 0$ for all $\{i,j\} \in S$.
2. For each iteration ($l = 0, 1, 2, \ldots$):

   (a) For each row X_i of $X^{(l)}$ ($i = 1, \ldots, n$):

      i. Compute $B_i = \sum_{j \neq i} W_{ij} X_j^T X_j$, where X_j is the jth row of $X^{(l)}$.
      ii. Compute λ_i as the largest eigenvalue of B_i.
      iii. Compute $z = \lambda_i X_i - X_i B_i^T + \sum_{j \neq i} W_{ij} R_{ij} X_j$.
      iv. Compute S_i as the set of all $j < i$ such that $\{i,j\} \in S$. Denote this set as $S_i = \{S_i(1) \cdots S_i(m)\}$.
      v. If S_i is empty and $z \neq 0$, then set $X_i = z / \| z \|_2$.
      vi. If S_i is not empty, then perform the following:

A. Concatenate the transposes of the X_j rows for which $j \in S_i$ into a matrix $\widetilde{X}$. That is,
$$\widetilde{X} = \begin{bmatrix} X^T_{S_i(1)} & \cdots & X^T_{S_i(m)} \end{bmatrix}.$$
B. Find a set of linearly independent row vectors that form a basis for the subspace that is orthogonal to the columns of $\widetilde{X}$. (This can be performed, for example, using QR decomposition.) Denote these basis vectors as $\{v_k\}$.
C. Compute $q = \sum_k (v_k z) v_k / (v_k v_k^T)$.
D. If $q \neq 0$, then set $X_i = q / \| q \|_2$.

(b) Set $X^{(l+1)} = \begin{bmatrix} X^T_1 & \cdots & X^T_n \end{bmatrix}^T$.

(c) Check for a termination criterion. This can be done, for example, by limiting the maximum iteration count l, or checking how much $X^{(l)}$ has changed from the previous iteration.

3. Feasibility and convergence issues

Now we consider the feasibility of the correlation approximation problem, and the ability of the majorization algorithm to find a feasible solution. X is an $n \times d$ matrix, so depending on the value of d and the set S, (4) may not have a solution. As a simple example, consider the 2×2 matrix $R = I$. Suppose that we want to use $d = 1$ and find a 2×1 matrix X such that $\widehat{R} = XX^T$ is a correlation matrix that minimizes $f(X)$ such that $\widehat{R}_{12} = \widehat{R}_{21} = 0$. This means that $\widehat{R}$ must be full rank, but since X has a rank of 1, XX^T also has a rank of 1. There is no 2×1 matrix X such that $\widehat{R} = XX^T$ is a correlation matrix satisfying the constraint $\widehat{R}_{12} = \widehat{R}_{21} = 0$.

The cost function $f(X)$ may actually increase from one iteration of Algorithm 1 to the next. The majorization algorithm guarantees a decrease in $f_i(X_i)$ at each iteration as shown in [26], but only if X_i is unconstrained with respect to previous rows of X; that is, only if there are no constraints $X_i X_j^T = 0$ for $j < i$.

As an example of how the cost function can increase from one iteration to the next, consider a 3×3 matrix X with the constraint $X_2 X_3^T = X_3 X_2^T = 0$. We begin the algorithm with

$$X^{(0)} = \begin{bmatrix} (X_1^{(0)})^T & (X_2^{(0)})^T & (X_3^{(0)})^T \end{bmatrix}^T. \tag{18}$$

At the first row iteration we find $X_1^{(1)}$ such that $f_1(X_1^{(1)}) < f_1(X_1^{(0)})$. At the second row iteration we find $X_2^{(1)}$ such that $f_2(X_2^{(1)}) < f_2(X_2^{(0)})$. But at the third row iteration we have to enforce the constraint $X_3 X_2^T = 0$. X_2 changed in the previous step, so the X_3 that we find may result in $f_3(X_3^{(1)}) > f_3(X_3^{(0)})$. This could then result in $f(X^{(1)}) > f(X^{(0)})$. This possibility is an unavoidable consequence of the row-wise minimization of $f(X)$, combined with the constraint $\widehat{R}_{ij} = 0$ for all $\{i, j\} \in S$.

When the optimization algorithm converges, it converges to a local minimum, not necessarily a global minimum. $f(X)$ in (3) is convex, and the constraint $X_i X_j^T = 0$ for all $(i, j) \in S$ is convex, but the constraint $X_i X_i^T = 1$ for all i is not convex. Therefore the constrained minimization problem is not convex and it may have multiple local minima. The limit of the sequence $f(X^{(l)})$ of the optimization algorithm will thus depend on the initial guess $X^{(0)}$.

In the derivation of the optimization algorithm, we used λ_i, the maximum eigenvalue of B_i, starting in (8). However, instead of using λ_i we could have used $\alpha \lambda_i$ for any $\alpha \geq 1$. As α increases the difference between $f_i(x)$ and $g_i(x)$ increases. This causes $g_i(x)$ to be steeper at the starting point of its minimization, which typically results in its constrained minimum being closer to its starting point, which results in more iterations before convergence, which increases computational effort. However, if an upper bound is used for λ_i rather than an exact calculation of λ_i [26], this could result in a net decrease of computational effort even though more iterations are required for convergence.

X_i is modified in Steps 2(a)v and 2(a)vi(D) of the algorithm, but only if z or q respectively are nonzero. If z or q are zero in those steps, then X_i should be set to any row vector with a norm of one.
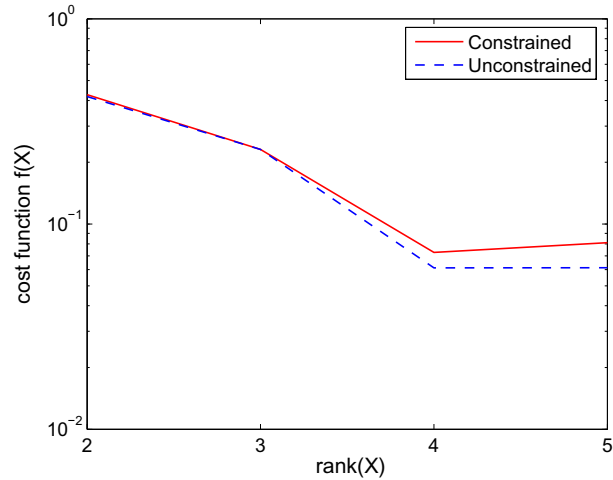
Fig. 2. Constrained and unconstrained majorization results for a 5×5 matrix.

The algorithm assumes that each row of X has unity norm. When the algorithm modifies X_i, it always does so in a way that maintains the unity norm of X_i. But if rows other than X_i do not have unity norm, then this assumption will be violated. Therefore the initial guess for X should be such that each row of X has unity norm.

Step 2(a) of the algorithm loops from $i = 1$ to n, where n is the number of rows in X. However, there is no reason why the row index needs to run from 1 to n in sequential order. The row index could just as well run from $p(1)$ to $p(n)$, where p is any permutation of the numbers $\{1, \ldots, n\}$. Since the algorithm converges to a local minimum of the objective function (rather than a global minimum), changes like this will, in general, result in different solutions at convergence. This has been discussed in [26].

W_{ij} for $(i, j) \in S$ does not affect the minimum of $f(X)$, but it does affect the majorization algorithm. So the values of W_{ij} for $(i, j) \in S$ can be considered as tuning parameters that can take any nonnegative values.

4. Numerical results

The unconstrained majorization approach to correlation matrix approximation has been compared in [26] to the Lagrange multiplier approach [30], geometric programming [15], and parameterization [28]. It was concluded in [26] that majorization was the most efficient of the four algorithms. That is, on average, the majorization approach converged to the best solution within a given CPU time. In this section we explore the use of majorization with and without the $\widehat{R}_{ij} = 0$ constraints. We terminated the majorization iterations when the cost function decreased by less than 0.01% from one iteration to the next, or when the cost function decreased to less than 0.0001. The initial guess for X was obtained using PCA as defined in (1).

The first matrix we consider is from [1].

$$
R = \begin{bmatrix}
1 & 0.5 & 0.5 & 0 & 0 \\
0.5 & 1 & 0.8 & 0.8 & 0.8 \\
0.5 & 0.8 & 1 & 0.8 & 0.8 \\
0 & 0.8 & 0.8 & 1 & 0.8 \\
0 & 0.8 & 0.8 & 0.8 & 1
\end{bmatrix}
\tag{19}
$$

This matrix has one negative eigenvalue and so it is not a correlation matrix. We can use constrained and unconstrained majorization to find an $\widehat{R}$ of a given rank d that is close to R. Constrained majorization has two $\widehat{R}_{ij} = 0$ constraints. Fig. 2 shows the cost function obtained with constrained majorization and unconstrained majorization for various values of d, which is the number of columns of X, and which is
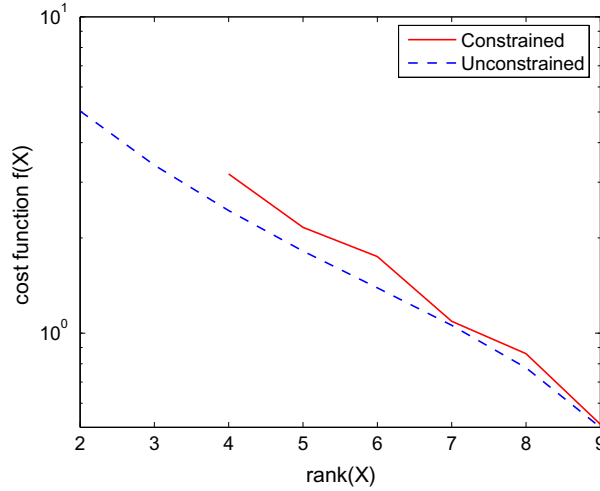
Fig. 3. Constrained and unconstrained majorization results for a 10×10 matrix. Constrained majorization requires rank (X) 4 because of the arrangement of the constraints. Both cost functions for rank $(X) = 10$ are zero (since R is a valid full rank correlation matrix) and so are not shown in the figure.

also the rank of $\widehat{R}$. Note that the constrained cost function increases slightly from $d = 4$ to $d = 5$, even though theoretically the global minimum of $f(X)$ when $d = 5$ is clearly less than or equal to the global minimum when $d = 4$. This illustrates the fact that the majorization algorithm does not necessarily find a global minimum of the cost function.

The second matrix we consider is based on [6, Section 6.9]. We create a 10×10 matrix R such that

$$R_{ij} = \begin{cases} \exp(-|i - j|) & \text{if } \exp(-|i - j|) > 0.001, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

This is the type of matrix that arises in financial applications, and it results in a valid correlation matrix. This matrix has six correlations that are zero. Fig. 3 shows the cost function obtained with constrained majorization and unconstrained majorization for various values of d. Constrained majorization requires d 4 because of the arrangement of the $\widehat{R}_{ij} = 0$ constraints. The last row of R has three elements $R_{ij} = 0$ for $j < i$, which means that d 4 for a solution to exist, as discussed in the first paragraph of Section 3.

The third matrix we consider is a 12×12 matrix from a financial example [28], where we have replaced with zeros all elements whose magnitudes are less than 0.01. This matrix has two negative eigenvalues and so it is not a correlation matrix. This matrix has six correlations that are zero. Fig. 4 shows the cost function obtained with constrained majorization and unconstrained majorization for various values of d. Constrained majorization requires d 3 because of the arrangement of $\widehat{R}_{ij} = 0$ constraints. The 8th, 10th, and 12th rows of R each have two elements $R_{ij} = 0$ for $j < i$, which means that d 3 for a solution to exist, as discussed in the first paragraph of Section 3.

The fourth matrix we consider is a 100×100 random correlation matrix generated with Matlab's "gallery" function, where we have replaced with zeros all elements whose magnitudes are less than 0.01. The matrix is a valid full-rank correlation matrix and has 1434 correlations that are zero. Fig. 5 shows the cost function obtained with constrained majorization and unconstrained majorization for various values of d. Constrained majorization requires d 18 because of the arrangement of $\widehat{R}_{ij} = 0$ constraints.

The fifth matrix we consider is a 31×31 correlation matrix defined from a House of Quality for a refrigerator product plan [19]. The rows and columns in the matrix define correlations between 31 metrics such as compressor efficiency, freezer shelf height, and warranty period. The matrix is extremely sparse, with only 66 off-diagonal nonzero correlations and 864 zero correlations. The matrix is indefinite with 29 positive and two negative eigenvalues. Fig. 6 shows the cost function obtained
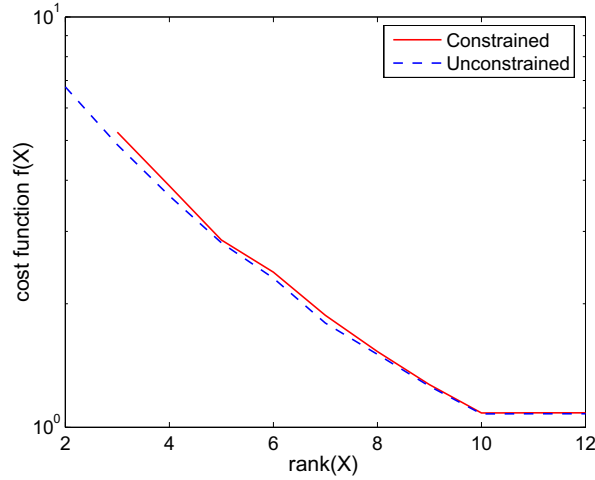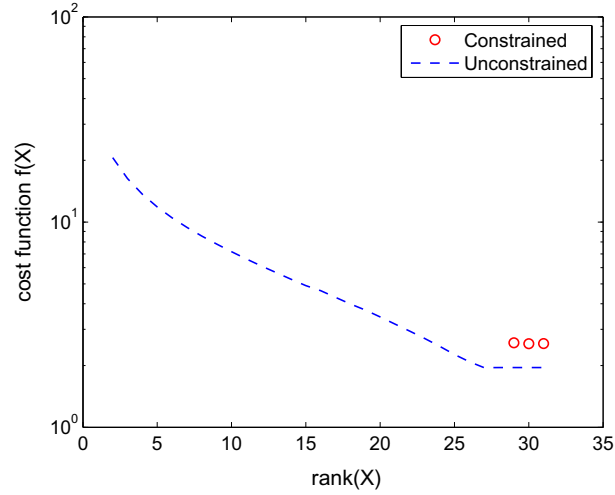
Fig. 4. Constrained and unconstrained majorization results for a 12×12 matrix. Constrained majorization requires rank $(X) \geq 3$ because of the arrangement of the constraints.
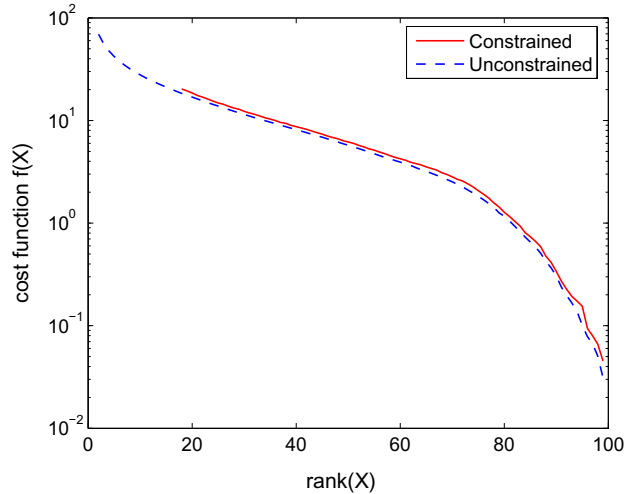
Fig. 5. Constrained and unconstrained majorization results for a 100×100 matrix. Constrained majorization requires rank $(X) \geq 18$ because of the arrangement of the constraints. Both cost functions for rank $(X) = 100$ are zero (since R is a valid full rank correlation matrix) and so are not shown in the figure.

with constrained majorization and unconstrained majorization for various values of d. Constrained majorization requires $d \geq 29$ because there are so many $\widehat{R}_{ij} = 0$ constraints.

Computational effort per iteration is greater with constrained than unconstrained majorization. Tables 1 and 2 show the CPU effort required for unconstrained and constrained majorization for the five test matrices discussed in this section. Table 1 shows the results when $\widehat{R}$ has the largest possible rank (5, 10, 12, 31, and 100, respectively). Table 2 shows the results when $\widehat{R}$ has the smallest possible rank (2, 4, 3, 29, and 18, respectively). We see that constrained majorization can take significantly longer per iteration than unconstrained majorization. This is due to the QR decomposition in Step 2(a)vi(B) of the algorithm. However, we also see that when the rank of $\widehat{R}$ is small as in Table 2, constrained majorization can take significantly fewer iterations to converge than unconstrained majorization. This is because the combination of constraints and low-rank $\widehat{R}$ results in fewer degrees of freedom in the optimization

Fig. 6. Constrained and unconstrained majorization results for a 31 × 31 matrix. Constrained majorization requires rank (X) 29 because of the constraints.

Table 1

Number of iterations and CPU time (s) for unconstrained and constrained majorization for five test matrices. The rank of the approximating correlation matrix $\widehat{R}$ was specified to be the same as the dimension of the input matrix R.

dim(R)	Unconstrained			Constrained		
	#iter	CPU	CPU/iter	#iter	CPU	CPU/iter
5 × 5	124	0.48	0.0039	177	4.01	0.0227
10 × 10	70	1.04	0.0149	73	1.80	0.0247
12 × 12	20	0.43	0.0215	20	0.86	0.0430
31 × 31	19	2.42	0.1274	2	1.09	0.5450
100 × 100	183	1618	8.8415	187	2229	11.9198

Table 2

Number of iterations and CPU time (s) for unconstrained and constrained majorization for five test matrices. The rank of the approximating correlation matrix $\widehat{R}$ was specified to be the minimum possible value while still allowing for a feasible constrained solution (rank($\widehat{R}$) = 2, 4, 3, 29, and 18, respectively).

dim(R)	Unconstrained			Constrained		
	#iter	CPU	CPU/iter	#iter	CPU	CPU/iter
5 × 5	8	0.07	0.0088	8	0.22	0.0275
10 × 10	24	0.43	0.0179	9	0.43	0.0478
12 × 12	18	0.44	0.0244	6	0.35	0.0583
31 × 31	17	2.24	0.1318	3	1.60	0.5333
100 × 100	27	26.26	0.9726	10	22.60	2.2600

problem. This can also be seen in the 31 × 31 matrix in Table 1, which has many constraints and therefore converges in only two iterations with constrained majorization.

5. Conclusion

We have derived a majorization approach for finding the closest correlation matrix $\widehat{R}$ of a user-specified rank to a given input matrix R, where "closest" is defined in terms of the weighted Frobenius norm of the error, and user-specified elements of $\widehat{R}$ are constrained to be zero. This is the first time that this particular problem has been studied. If there are no constraints then our method reduces to that proposed in [26] and convergence is guaranteed. If constraints are specified in the problem

then we cannot guarantee convergence, or even feasibility, depending on the constraints and the rank of $\widehat{R}$. However, for the sample problems explored in this paper, performance of the algorithm was good, with only a slight loss of estimation accuracy for the constrained problem compared to the unconstrained problem. Matlab source code for the algorithm presented in this paper can be downloaded from http://academic.csuohio.edu/simond/corr.

Future work along these lines could explore the effect of the W_{ij} weights for those values of i and j for which the constraint $\widehat{R}_{ij} = 0$ holds. These values of W_{ij} do not affect the minimum of the objective function, but they do affect the progress of the majorization algorithm and thus can be considered as tuning parameters. Another important area for future work is modifying the algorithm to guarantee convergence, or finding conditions under which convergence can be guaranteed.

One of the primary considerations of the majorization algorithm for real applications might be its large computational effort. There are two main sources of computational effort in the algorithm. The first is the solution of the largest eigenvalue of B_i. The second is the solution of a set of linearly independent basis vectors for the orthogonal complement of $\widetilde{X}$, which can be accomplished with QR factorization. The first problem could be solved by using an upper bound for λ_i in the majorization algorithm. One easy but conservative upper bound is $n - 1$ due to the unit length bound of the columns of $\widetilde{X}$ [26]. This would remove the $O(n^3)$ eigenvalue calculation from the majorization algorithm. The $O(n^3)$ effort of the QR factorization step of the majorization algorithm could be alleviated with one of the fast $O(n)$ QR algorithms that have been proposed in the literature [3].

References

[1] A. Al-Subaihi, Simulating correlated multivariate pseudorandom numbers, J. Statist. Software 9 (4) (2004) <www.jstatsoft.org/v09/i04>.
[2] M. Anjos, N. Higham, P. Takouda, H. Wolkowicz, A semidefinite programming approach for the nearest correlation matrix problem, Optimization Online, September 16, 2003. <www.optimization-online.org/DB_HTML/2003/09/729.html>.
[3] J. Apolinario, P. Diniz, A new fast QR algorithm based on a priori errors, IEEE Signal Process. Lett. 4 (11) (1997) 307–309.
[4] D. Bernstein, Matrix Mathematics, Princeton University Press, 2005.
[5] I. Borg, P. Groenen, Modern Multidimensional Scaling, Springer, 1997.
[6] D. Brigo, F. Mercurio, Interest Rate Models – Theory and Practice, Springer, 2006.
[7] D. Brigo, F. Mercurio, Calibrating LIBOR, Risk Mag. 15 (1) (2002) 117–122.
[8] C. Brooks, B. Scott-Quinn, J. Walmsley, Adjusting VAR models for the impact of the EURO, July 20, 1998.
[9] M. Budden, P. Hadavas, L. Hoffman, C. Pretz, Generating valid 4×4 correlation matrices, Appl. Math. E-Notes 7 (2007) 53–59.
[10] L. Cohen, Quality Function Deployment, Springer, 1995.
[11] J. Dash, Quantitative Finance and Risk Management, World Scientific, 2004.
[12] J. de Leeuw, A decomposition method for weighted least squares low-rank approximation of symmetric matrices, UCLA Department of Statistics Paper 2006041602, April 16, 2006. <http://repositories.cdlib.org/uclastat/papers/2006041602>.
[13] B. Flury, Common Principal Components and Related Multivariate Models, John Wiley & Sons, 1988.
[14] I. Grubisic, Interest Rate Theory: The BGM Model, Masters Thesis, Leiden University, August 2002. <www.math.uu.nl/people/grubisic>.
[15] I. Grubisic, R. Pietersz, Efficient rank reduction of correlation matrices, Linear Algebra Appl. 422 (2–3) (2007) 629–653.
[16] N. Higham, Matrix nearness problems and applications, in: M. Gover, S. Barnett (Eds.), Applications of Matrix Theory, Clarendon Press, 1989, pp. 1–27.
[17] N. Higham, Computing the nearest correlation matrix – A problem from finance, IMA J. Numer. Anal. 22 (3) (2002) 329–343.
[18] W. Hoge, A subspace identification extension to the phase correlation method, IEEE Trans. Med. Imaging 22 (2) (2003) 277–280.
[19] Kenneth Crow and Associates, Refrigerator Product Planning Matrix, 2007. <http://www.npd-solutions.com/refrigppm.html>.
[20] T. Kourti, Process analysis and abnormal situation detection: from theory to practice, IEEE Control Syst. Mag. 22 (5) (2002) 10–25.
[21] S. Mishra, Optimal solution of the nearest correlation matrix problem by minimization of the maximum norm, Social Sci. Res. Network, August 6, 2004. <http://ssrn.com/abstract=573241>.
[22] S. Mishra, Completing correlation matrices of arbitrary order by differential evolution method of global optimization: a Fortran program, Social Sci. Res. Network, March 5, 2007. SSRN: <http://ssrn.com/abstract=968373>.
[23] S. Mishra, The nearest correlation matrix problem: solution by differential evolution method of global optimization, Research Papers in Economics, April 14, 2007. <http://ideas.repec.org/p/pra/mprapa/2760.html>.
[24] T. Moon, W. Stirling, Mathematical Methods and Algorithms for Signal Processing, Prentice-Hall, 2000.
[25] M. Morini, N. Webber, An EZI method to reduce the rank of a correlation matrix in financial modelling, Appl. Math. Finance 13 (4) (2006) 309–331.
[26] R. Pietersz, P. Groenen, Rank reduction of correlation matrices by majorization, Quant. Finance 4 (6) (2004) 649–662.
[27] H. Qi, D. Sun, Correlation stress testing for value-at-risk: an unconstrained convex optimization approach, March 2007. <http://www.math.nus.edu.sg/~matsundf>.

[28] R. Rebonato, P. Jackel, The most general methodology to create a valid correlation matrix for risk management and option pricing purposes, J. Risk 2 (2) (1999) 17–27.

[29] D. Simon, Reduced order Kalman filtering without model reduction, Control Intell. Syst. 35 (2) (2007) 169–174.

[30] Z. Zhang, L. Wu, Optimal low-rank approximation to a correlation matrix, Linear Algebra Appl. 364 (2003) 161–187.