Pergamon

Computers Math. Applic. Vol. 35, No. 6, pp. 73–79, 1998

0898-1221/98 $19.00 + 0.00

PII: S0898-1221(98)00020-0

Dykstra's Algorithm for Constrained Least-Squares Rectangular Matrix Problems

R. ESCALANTE AND M. RAYDAN
Depto. de Computación, Facultad de Ciencias
Universidad Central de Venezuela
Ap. 47002, Caracas 1041-A, Venezuela
rescalan@kuaimare.ciens.ucv.ve mraydan@reacciun.ve

(Received and accepted June 1997)

Abstract—In a recent paper, the authors applied Dykstra's alternating projection algorithm to solve constrained least-squares $n \times n$ matrix problems. We extend these results in two different directions. First, we make use of the singular value decomposition to solve now constrained least-squares rectangular $m \times n$ matrix problems that arise in several applications. Second, we propose a new and improved implementation of the projection algorithm onto the ϵ-positive definite set of matrices. This implementation does not require the computation of all eigenvalues and eigenvectors of a matrix per iteration, and still guarantees convergence. Finally, encouraging preliminary numerical results are discussed.

Keywords—Alternating projection methods, Dykstra's algorithm, Singular value descomposition, Constrained least-squares, Gerschgorin circles.

1. INTRODUCTION

We are interested in solving constrained least-squares *rectangular* matrix problems. More precisely, we consider the following constrained optimization problem:

$$\min \|AX - B\|_F^2, \tag{1}$$

subject to

$$X^T = X, \tag{2}$$

$$L \leq X \leq U, \tag{3}$$

$$\lambda_{\min}(X) \geq 0, \tag{4}$$

where A and B are given $m \times n$ real matrices, $m \geq n$, $\operatorname{rank}(A) = n$, and X is the symmetric $n \times n$ matrix that we wish to find. Likewise, L and U are given $n \times n$ real matrices, and $\lambda_{\min}(X)$ represents the smallest eigenvalue of X. Throughout this paper, the notation $A \leq B$, for any two

The authors were partially supported by the Parallel and Distributed Computing Center at UCV.
The authors are grateful to Y. Alberti and D. Duran from Universidad Central de Venezuela for programming assistance.

real $n \times n$ matrices, means that $A_{ij} \leq B_{ij}$ for all $1 \leq i,\ j \leq n$. Also, $\|A\|_F$ denotes the Frobenius norm of a real matrix A, defined as

$$\|A\|_F^2 = \langle A, A \rangle = \sum_{i,j=1}^{n} (A_{ij})^2,$$

where the inner product is given by $\langle A, B \rangle = \text{trace}\,(A^T B)$.

Problems (1)–(4) arise naturally in statistics and mathematical economics. Simplified versions of problems (1)–(4) have been recently studied. See, for instance, [1–7]. In all these cases, at least one constraint is not taken into account. However, for the general case, the problem has received little attention. To the best of our knowledge, problems (1)–(4) have been only considered by Hu [8] and more recently by Escalante and Raydan [9]. In the foregoing case $m = n$, A is the identity matrix, X is also constrained to have a particular linear pattern $\mathcal{P}$, and $\lambda_{\min}(X) \geq \varepsilon > 0$. In our extension to the rectangular case, and for the sake of clarity, we consider $\mathcal{P}$ as a free symmetric linear pattern and $\varepsilon = 0$.

We solve problems (1)–(4) by applying once again Dykstra's alternating projection method (see [10,11]). Although now, our approach is based on the use of the singular value decomposition (SVD) to transform the original problem into a simpler one that fits nicely with the algorithms developed in [9]. For a recent application of Dykstra's algorithm, see [12], and for complete discussions on alternating projection methods, see [13,14].

In this work, we also propose a new and improved implementation of the projection algorithm onto the ε-positive definite set of matrices. This implementation does not require the computation of all eigenvalues and eigenvectors of a matrix per iteration, as the original one proposed by Escalante and Raydan [9], and still guarantees convergence.

The rest of the paper is organized as follows. In Section 2, we carry out the definition of the feasible region to solve the rectangular case. We make extensive use of the SVD to obtain the rectangular version of the alternating projection algorithm. Then, in Section 3, we propose the new implementation of the projection algorithm onto the ε-positive definite set of matrices. Finally, in Section 4, we present preliminary numerical results.

2. THE RECTANGULAR CASE

The feasible region of problems (1)–(4) are the intersection of two sets denoted by $\mathcal{B}\text{ox}\,(\mathcal{B})$ and positive semidefinite (psd), given by

$$\mathcal{B} = \{X \in \mathbb{R}^{n\times n} : L \leq X \leq U\}, \qquad \text{and}$$
$$\text{psd} = \{X \in \mathbb{R}^{n\times n} : X^T = X,\ \lambda_{\min}(X) \geq 0\}.$$

Problems (1)–(4) can be stated as follows:

$$\min\left\{\|AX - B\|_F^2 : X \in \mathcal{B} \cap \text{psd}\right\}. \tag{5}$$

We observe that the feasible region of problem (5) is the intersection of closed and convex sets in the inner product space $\mathbb{R}^{n\times n}$, and so it is also a closed and convex set.

The authors in [9] solve (5), when $m = n$ and $A = I$, by means of a modification of the Dykstra's alternating projection method [10,11]. More explicitly, they solve this problem by projecting cyclically onto every one of the closed and convex sets whose intersection is the feasible region of (5). They present two alternating projection algorithms [9, Algorithms (3.5) and (4.1)], and characterize the projection onto every one of the sets involved in the algorithms, and use the convergence result established by Boyle and Dykstra [10, Theorem 2], to obtain convergence in the Frobenius norm to the unique solution of problem (5).

We will now use the same strategy to solve problem (5). In particular, we use the idea of Higham in [5], where he applies the SVD to analyse the "symmetric Procrustes" problem, see also [2]. This problem consists in finding the symmetric matrix X which minimizes the Frobenius (or Euclidean) norm of $AX - B$, where A and B are given rectangular matrices.

Let the matrix A in (5) have the following SVD:

$$A = P \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} Q^{\top},$$

where $P \in \mathbb{R}^{m \times m}$ and $Q \in \mathbb{R}^{n \times n}$ are orthogonal and

$$\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n), \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0.$$

Then, in accordance with [5], we have

$$\|AX - B\|_F^2 = \|\Sigma Y - C_1\|_F^2 + \|C_2\|_F^2,$$

where

$$Y = Q^{\top} X Q, \qquad C = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = P^{\top} B Q, \qquad C_1 = (C_{ij}) \in \mathbb{R}^{n \times n}.$$

Thus, the problem reduces to minimizing the quantity $\|\Sigma Y - C_1\|_F^2$. If X is constrained to be symmetric, then the matrix Y will be also symmetric, and the solution for $1 \leq i,\ j \leq n$ is given by

$$y_{ij} = \begin{cases} \dfrac{(\sigma_i c_{ij} + \sigma_j c_{ji})}{(\sigma_i^2 + \sigma_j^2)}, & \sigma_i^2 + \sigma_j^2 \neq 0, \\ \text{arbitrary}, & \text{otherwise.} \end{cases} \tag{6}$$

Finally, the required solution X is given by $X = QYQ^{\top}$. See [5] for details.

In order to study the transformed feasible region, we now define two sets in $\mathbb{R}^{n \times n}$. The first set is denoted by psd$'$, and is given by

$$\text{psd}' = \left\{ Z \in \mathbb{R}^{n \times n} : Z = \Sigma Y,\ Y^{\top} = Y,\ \lambda_{\min}(Z) \geq 0 \right\},$$

where $Y = Q^{\top} X Q$. Since $X \in$ psd, then $Y \in$ psd, and we observe that $\lambda_{\min}(\Sigma Y) \geq 0$ if and only if $\lambda_{\min}(Y) \geq 0$. Therefore, minimizing $\|\Sigma Y - C_1\|_F$, over all $Y \in$ psd, is equivalent to solving the problem

$$\min_{Z \in \text{psd}'} \|Z - C_1\|_F. \tag{7}$$

The second set corresponds to $\mathcal{B}$ and is given by

$$\mathcal{B}' = \left\{ Z \in \mathbb{R}^{n \times n} : L \leq Q \Sigma^{-1} Z Q^{\top} \leq U \right\}.$$

It is also clear that minimizing $\|\Sigma Q^{\top} X Q - C_1\|_F$, over all $X \in \mathcal{B}$ is equivalent to solving the problem

$$\min_{Z \in \mathcal{B}'} \|Z - C_1\|_F. \tag{8}$$

Consequently, using (7) and (8), problem (5) can now be stated as follows:

$$\min \left\{ \|Z - C_1\|_F^2 : Z \in \text{psd}' \cap \mathcal{B}' \right\}. \tag{9}$$

We observe that the feasible region of problem (9) is the intersection of closed and convex sets in $\mathbb{R}^{n \times n}$. As in [9], we denote by $P_{\text{psd}'}$ and $P_{\mathcal{B}'}$ the projections onto psd$'$ and $\mathcal{B}'$, respectively.

We now present our version of Dykstra's alternating projection algorithm to solve problem (9).

Given $C_1 \in \mathbb{R}^{n\times n}$, set $(C_1)_0 = C_1$ and $I^0_{\text{psd}'} = I^0_{\mathcal{B}'} = 0$.

For $i = 0, 1, 2, \ldots,$

$$\begin{aligned} (C_1)_i &= P_{\mathcal{B}'}(C_1)_i - I^i_{\text{psd}'}, \\ I^{i+1}_{\text{psd}'} &= P_{\text{psd}'}(C_1)_i - (C_1)_i, \\ (C_1)_{i+1} &= P_{\text{psd}'}(C_1)_i - I^i_{\mathcal{B}'}, \\ I^{i+1}_{\mathcal{B}'} &= P_{\mathcal{B}'}(C_1)_{i+1} - (C_1)_{i+1}. \end{aligned} \tag{10}$$

Here $I^i_{\text{psd}'}$ and $I^i_{\mathcal{B}'}$ play the role of the increments introduced by Dykstra [11]. Using [10], we obtain the following convergence result for algorithm (10).

THEOREM 2.1. *If the closed and convex set* $\text{psd}' \cap \mathcal{B}'$ *is not empty, then for any* $C_1 \in \mathbb{R}^{n\times n}$ *the sequences* $\{P_{\text{psd}'}(C_1)_i\}$ *and* $\{P_{\mathcal{B}'}(C_1)_i\}$, *generated by algorithm (10), converge in the Frobenius norm to the unique solution of problem (9).*

REMARKS.

(1) To project C_1 onto $\mathcal{B}'$, i.e., to solve problem (8), we first project the matrix $Q\Sigma^{-1}C_1Q^\top$ onto the box $\mathcal{B}$ to obtain a matrix $X \in [L, U]$ (see [9] for details). Then, $P_{\mathcal{B}'}(C_1) = \Sigma Q^\top X Q$.

(2) To project C_1 onto psd' (problem (7)), we first compute Y using (6) and then project Y onto the set psd. Finally, $P_{\text{psd}'}(C_1) = \Sigma P_{\text{psd}}(Y)$. To compute $P_{\text{psd}}(Y)$, we can use the procedure described in Section 3 with $\varepsilon = 0$.

(3) We have only considered the linear pattern $\mathcal{P}$, defined in [9], as a free symmetric pattern. We observe that a more constrained pattern would involve a much more complicated treatment. Moreover, we believe that each particular pattern (for example, Toeplitz, Hankel, etc.) should be study individually to obtain an expression similar to (6).

3. PROJECTION ONTO THE SET ε-pd

We now propose a new and improved implementation of the projection algorithm onto the ε-positive definite set of matrices ($\varepsilon \geq 0$). This implementation does not require the computation of all eigenvalues and eigenvectors of a matrix per iteration, and still guarantees convergence.

In [9, Section 3], the authors characterize the projection onto the set $\varepsilon\,\text{pd}$. They also discuss an implementation that requires the computation of all eigenvalues and eigenvectors of a symmetric matrix X^k at every cycle k. Our improved implementation is given by

$$P_{\varepsilon\,\text{pd}}\left(X^k\right) = X^k + \sum_{i=1}^{p} \left(\varepsilon - {\lambda_i}^k\right) {Z_i}^k {Z_i}^{k\top}, \tag{11}$$

where X^k is the matrix obtained at the k^{th} cycle by projecting onto the set $\mathcal{B}\text{ox} \cap \mathcal{P}$, which in turn will be projected onto the set $\varepsilon\,\text{pd}$. Likewise, $\lambda_1, \ldots, \lambda_p$ are the eigenvalues of X^k that are less than ε, and ${Z_1}^k, \ldots, {Z_p}^k$ are the corresponding eigenvectors.

Our next result establishes the equivalence between (11) and the procedure described in [9, Section 3].

THEOREM 3.1. *The projection* $P_{\varepsilon\,\text{pd}}(X^k)$ *onto the set* ε *pd is given by (11).*

PROOF. Consider the following spectral decomposition of X^k:

$$X^k = \lambda_1^k Z_1^k Z_1^{k\top} + \cdots + \lambda_p^k Z_p^k Z_p^{k\top} + \cdots + \lambda_n^k Z_n^k Z_n^{k\top}, \tag{12}$$

where $\lambda_1^k, \ldots, \lambda_n^k$ are the eigenvalues of X^k, and $Z_1, \ldots, Z_n$ are corresponding eigenvectors. If we project onto the set $\varepsilon\,\mathrm{pd}$, we obtain (see [9, Section 3])

$$P_{\varepsilon\,\mathrm{pd}}\left(X^k\right) = \varepsilon Z_1^k Z_1^{k\top} + \cdots + \varepsilon Z_p^k Z_p^{k\top} + \sum_{i=p+1}^{n} \lambda_i^k Z_i^k Z_i^{k\top}. \tag{13}$$

Notice that, using (12), the second term in the right-hand side of (13) can be written as

$$\sum_{i=p+1}^{n} \lambda_i^k Z_i^k Z_i^{k\top} = X^k - \lambda_1^k Z_1^k Z_1^{k\top} - \cdots - \lambda_p^k Z_p^k Z_p^{k\top},$$

and replacing this expression back in (13), we obtain

$$P_{\varepsilon\mathrm{pd}}\left(X^k\right) = \varepsilon Z_1^k Z_1^{k\top} + \varepsilon Z_2^k Z_2^{k\top} + \cdots + \varepsilon Z_p^k Z_p^{k\top} + X^k - \lambda_1^k Z_1^k Z_1^{k\top} - \lambda_2^k Z_2^k Z_2^{k\top} - \cdots - \lambda_p^k Z_p^k Z_p^{k\top}.$$

Therefore,

$$P_{\varepsilon\mathrm{pd}}\left(X^k\right) = X^k + \sum_{i=1}^{p} \left(\varepsilon - \lambda_i^k\right) Z_i^k Z_i^{k\top},$$

and the result is established. ∎

REMARKS.

(1) In most real applications, p becomes "small" in the first few cycles, and remains constant during the rest of the process. Hence, as we will show in Section 4, the CPU time is significantly reduced when (11) is used.

(2) In our implementation, we use the following safeguard technique. In the first cycle, we use the subroutine DSYEVX from the LAPACK library to calculate the eigenvalues (and their corresponding eigenvectors) in the interval $[lb, \varepsilon]$. We compute the lower bound lb by applying the Gerschgorin circle theorem [15, Theorem 7.2.1].

From that point on, we use the subroutine DSYEVX to calculate the $p+1$ smallest eigenvalues of X^k, where p is the number of eigenvalues that were less than ε in the previous cycle. If $\lambda_{p+1} < \varepsilon$, which happens very seldom, then the number of eigenvalues that are less than ε has increased, and we update p using once again the Gerschgorin circle theorem. If $\lambda_{p+1} \geq \varepsilon$, then the process continues with p as in the previous cycle or even smaller.

4. NUMERICAL RESULTS

We compare Algorithm 4.1 in [9] with the original implementation (OLD for brevity), and the new implementation (NEW) given by the modified projection (11). In particular, we are interested in the computational work required in both cases for "small" values of p ($p \ll n$), and also for "large" values of p ($p \approx n$).

All experiments in this section were run on a HP APOLLO 735 workstation in double precision FORTRAN. The eigenvalues and eigenvectors, required in the projection onto the set $\varepsilon\,\mathrm{pd}$, were computed by either subroutine DSYEV or DSYEVX from the LAPACK library [16]. We set $\varepsilon = 0.1$, for all experiments. The iterations in both cases were stopped when

$$\left\|P_{\varepsilon\,\mathrm{pd}}(A_{i+1}) - P_{\varepsilon\,\mathrm{pd}}(A_i)\right\|_F \leq \mathrm{TOL},$$

for different values of TOL.

In [9], we use the subroutine DSYEV to compute all eigenvalues and eigenvectors of a given matrix A. In this work, we also use the subroutine DSYEVX that provides flexibility to compute selected eigenvalues and eigenvectors.

The tables below show the dimension of the problem (n), the CPU time in seconds (TIME), the number of iterations (IT) required by both implementations for different tolerances, and the distance (ERROR) in the Frobenius norm between the output matrices and the exact solution X.

Experiment 1

In our first experiment, we use the initial matrix A whose ij-entry is given by

$$A_{ij} = \begin{cases} \left(\dfrac{i}{i+j-1}\right) + 0.1, & \text{if } i = j, \\ \dfrac{i}{i+j-1}, & \text{if } i \neq j, \end{cases}$$

for $i = 1, \ldots, n-1$, $j = 1, \ldots, n$, and $A_{ij} = 0.01$, for $i = n$, $j = 1, \ldots, n$. We also define the matrix L as the null matrix and U as the matrix whose ij-entry is given by $i + j$. Last, let the pattern matrix be a symmetric Toeplitz matrix (see [9, Experiment 2]). In this case, problem (5) is feasible.

The matrix A has $n/2$ eigenvalues smaller than ε (first cycle), and few (one or two) during the rest of the process. Table 1 shows the obtained numerical results.

Table 1. Results for Experiment 1.

		OLD			NEW		
n	TOL	IT	TIME	ERROR	IT	TIME	ERROR
10	10^{-2}	7	6.0×10^{-2}	3.11×10^{-3}	7	4.0×10^{-2}	3.11×10^{-3}
	10^{-5}	17	0.11	2.59×10^{-6}	17	9.0×10^{-2}	2.59×10^{-6}
	10^{-7}	24	0.22	4.0×10^{-8}	24	0.13	3.98×10^{-8}
100	10^{-2}	10	48.39	4.69×10^{-3}	10	11.15	4.69×10^{-3}
	10^{-5}	48	219.83	4.78×10^{-6}	48	34.55	4.78×10^{-6}
	10^{-7}	79	363.08	4.58×10^{-8}	79	53.61	4.58×10^{-8}

Table 1 shows clearly that *NEW* is faster than *OLD*. Specifically, we see that as the tolerance decreases the difference increases. Moreover, if n increases, the CPU time for *NEW* is significantly less than the CPU time for *OLD* to attain the same accuracy.

Experiment 2

The initial matrix A used in the second experiment has $n-1$ eigenvalues smaller than ε (first cycle), and $n-2$ during the rest of the process. The matrix A is defined as follows:

$$A_{ij} = \frac{1}{ij} + (i - j), \qquad i, j = 1, \ldots, n,$$

where $ij = i + j - 1$. Here the matrices L and U, and the set $\mathcal{P}$, are as in Experiment 1. Table 2 shows the numerical results for this experiment.

Table 2. Results for Experiment 2.

		OLD			NEW		
n	TOL	IT	TIME	ERROR	IT	TIME	ERROR
10	10^{-2}	3	0.02	4.0×10^{-3}	3	3.0×10^{-2}	4.03×10^{-3}
	10^{-5}	61	0.4	4.7×10^{-6}	61	0.6	4.77×10^{-6}
	10^{-7}	147	1.0	4.8×10^{-8}	147	1.41	4.89×10^{-8}
100	10^{-2}	3	14.4	4.7×10^{-3}	3	14.6	4.69×10^{-3}
	10^{-5}	77	364.5	4.8×10^{-6}	77	372.83	4.84×10^{-6}
	10^{-7}	322	1536.2	5.0×10^{-8}	322	1625.13	4.97×10^{-8}

In this case, we observe that *OLD* is slightly faster than *NEW*. This experiment shows the disadvantage of using the improved version when $p \approx n$. Indeed, subroutine DSYEVX from LAPACK seems to be an excellent choice to compute eigenvalues and eigenvectors when $p \ll n$. However, if p is "large", then computing all eigenvalues at every cycle by means of subroutine DSYEV is a better option.

REFERENCES

1. J.C. Allwright, Positive semidefinite matrices: Characterization via conical hulls and least-squares solution of a matrix equation, *SIAM J. Control and Opt.* **26**, 537–555, (1988).
2. L.E. Andersson and T. Elfving, A constrained Procrustes problem, *SIAM J. Matrix Anal. Appl.* **18**, 124–139, (1997).
3. T.L. Hayden and J. Wells, Approximation by matrices positive semidefinite on a subspace, *Linear Algebra Appl.* **109**, 115–130, (1988).
4. N.J. Higham, Computing a nearest symmetric positive semidefinite matrix, *Linear Algebra Appl.* **103**, 103–118, (1988).
5. N.J. Higham, The symmetric Procrustes problem, *BIT* **28**, 133–143, (1988).
6. H. Hu and I. Olkin, A numerical procedure for finding the positive definite matrix closest to a patterned matrix, *Statistics and Probability Letters* **12**, 511–515, (1991).
7. T.J. Suffridge and T.L. Hayden, Approximation by a Hermitian positive semidefinite Toeplitz matrix, *SIAM J. Matrix Anal. Appl.* **14**, 721–734, (1993).
8. H. Hu, Positive definite constrained least-squares estimation of matrices, *Linear Algebra Appl.* **229**, 167–174, (1995).
9. R. Escalante and M. Raydan, Dykstra's algorithm for a constrained least-squares matrix problem, *Num. Linear Algebra Appl.* **3** (6), 459–471, (1996).
10. J.P. Boyle and R.L. Dykstra, A method for finding projections onto the intersections of convex sets in Hilbert spaces, *Lecture Notes in Statistics* **37**, 28–47, (1986).
11. R.L. Dykstra, An algorithm for restricted least-squares regression, *J. Amer. Stat. Assoc.* **78**, 837–842, (1983).
12. W. Glunt, T.L. Hayden, S. Hong and J. Wells, An alternating projection algorithm for computing the nearest Euclidean distance matrix, *SIAM J. Matrix Anal. Appl.* **11**, 589–600, (1990).
13. H.H. Bauschke and J.M. Borwein, On projection algorithms for solving convex feasibility problems, *SIAM Review* **38**, 367–426, (1996).
14. F. Deutsch, The method of alternating orthogonal projections, In *Approximation Theory, Spline Functions and Applications*, (Edited by S.P. Singh), pp. 105–121, Kluwer Academic Publishers, Netherlands, (1992).
15. G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, (1989).
16. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen, *LAPACK User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, (1992).