Hybrid Methods for Minimizing Least Distance Functions with Semi-Definite Matrix Constraints

Suliman Al-Homidan
Department of Mathematics, King Fahd University of Petroleum and Minerals, Dhahran 31261, PO Box 119, Saudi Arabia

Abstract

Hybrid methods for minimizing least distance functions with semi-definite matrix constraints are considered. One approach is to formulate the problem as a constrained least distance problem in which the constraint is the intersection of three convex sets. The Dykstra-Han projection algorithm can then be used to solve the problem. This method is globally convergent but the rate of convergence is slow. However, the method does have the capability of determining the correct rank of the solution matrix, and this can be done in relatively few iterations. If the correct rank of the solution matrix is known, it is shown how to formulate the problem as a smooth nonlinear minimization problem, for which a rapid convergence can be obtained by l_1SQP method. Also this paper studies hybrid method that attempt to combine the best features of both types of methods. An important feature concerns the interfacing of the component methods. Thus, it has to be decided which method to use first, and when to switch between methods. Difficulties such as these are addressed in the paper. Comparative numerical results are also reported.

Key words : Alternating projections, least distance functions, non-smooth optimization, positive semi-definite matrix.
AMS (MOS) subject classifications 65F99, 99C25, 65F30

1 Introduction

Minimizing a general function subject to semi–definite matrix constraint is a problem which arises in many practical situations, particularly in statistics where the semi–definite matrix constraint is usually a covariance matrix with varying elements. In this paper a least distance problem of the following type is solved. Given a symmetric positive semi–definite matrix $F \in \mathbb{R}^{n\times n}$ then we consider

$$\begin{aligned} &\text{minimize} \quad \mathbf{x}^T\mathbf{x} \qquad \mathbf{x} \in \mathbb{R}^n \\ &\text{subject to} \quad \bar{F} + \operatorname{diag}\mathbf{x} \geq 0, \qquad \mathbf{x} \leq \mathbf{v} \end{aligned} \tag{1.1}$$

where $\operatorname{diag} \mathbf{v} = \operatorname{Diag} F$ and $\bar{F} = F - \operatorname{Diag} F$. This kind of problem is important by itself and it is also used subsequently in solving the educational testing problem [1]. Problem (1.1) can be more general if we express it as

$$\begin{aligned} &\text{minimize} \quad \|\mathbf{a} - \mathbf{x}\|_2^2 \qquad \mathbf{x} \in \mathbb{R}^n \\ &\text{subject to} \quad \bar{F} + \operatorname{diag} \mathbf{x} \geq 0, \qquad \mathbf{x} \leq \mathbf{v} \end{aligned} \tag{1.2}$$

where $\mathbf{a}$ is an initial point and then we have a different problem with every different $\mathbf{a}$. Problems of this type can be solved in a similar way to methods of this paper.

Two methods are developed for solving (1.1). Firstly, a projection algorithm is given for solving (1.1) which converges linearly or slower and globally. This method is described in Section 2. Secondly an implementation of the l_1 SQP method is used. Fletcher [1985] developed an algorithm for solving linear objective function with semi-definite matrix constraintsis. In Section 3 we follow his method but applyed to (1.1).

In Section 4 a hybrid method is described, which starts with the projection method to estimate the rank $r^{(k)}$ and continues with the l_1SQP method. Finally in Section 5 numerical comparisons of these methods are carried out. Hybrid methods have often been used successfully in optimization, (e.g. and Al-Homidan and Fletcher [2] and Al-Homidan [1]).

2 A Projection Algorithm

In this section a projection algorithm for solving (1.1) is described. The method described here depends on the basic iterated projection algorithm by [7].

It is convenient to define three convex sets for the purpose of constructing the probem. The set of all $n \times n$ symmetric positive semi–definite matrices

$$K_{\rm I\!R} \; = \; \{A : A \in {\rm I\!R}^{n\times n}, \;\; A^T \; = \; A \text{ and } \; \mathbf{z}^T A \mathbf{z} \; \geq \; 0 \qquad \forall \; \mathbf{z} \in {\rm I\!R}^n\} \tag{2.1}$$

is a convex cone of dimension $n(n+1)/2$. If $F \in {\rm I\!R}^{n\times n}$ is any given symmetric positive definite matrix, then define

$$K_{\rm off} \; = \; \{A : A \in {\rm I\!R}^{n\times n}, \; A \; - \; \text{Diag } A \; = \; \bar{F}\}. \tag{2.2}$$

This is the set of matrices whose off–diagonal elements are equal to those of F. Define

$$K_{\rm b} \; = \; \{A : A \in {\rm I\!R}^{n\times n}, A \; = \; \bar{A} \; + \; \text{diag } \mathbf{x}, \; x_i \; \leq \; v_i \;\; i = 1, 2, ...n\}. \tag{2.3}$$

This is the set of matrices that is obtained by reducing the diagonal of A. $K_{\rm off}$ and $K_{\rm b}$ are subspaces. Then (1.1) can be expressed as

$$\begin{aligned} &\text{minimize} \quad \|\bar{F} \; - \; A\| \\ &\text{subject to} \;\; A \; \in \; K_{\rm I\!R} \cap K_{off} \cap K_b. \end{aligned} \tag{2.4}$$

The matrix norm here means the Frobenius norm.

The projection on $K \; = \; \bigcap_{i=1}^{3} K_i$ is computed based on the Dykstra algorithm [3] given in Algorithm 2.1. It follows from [3] that the resulting method is globally convergent. (See also [5]).

Algorithm 2.1 Given any positive definite matrix F, let $F^{(0)} \; = \; F$

$$\begin{aligned} &For \quad k \; = \; 0, \; 1, \; 2, \; \ldots \\ &F^{(k+1)} \; = \; F^{(k)} \; + \; [P_b P_{off} P_{\rm I\!R}(F^{(k)}) \; - \; P_{\rm I\!R}(F^{(k)})] \end{aligned}$$

The projection map $P_{\mathbb{R}}(A)$ formula on to $K_{\mathbb{R}}$ is given by [6]

$$P_{\mathbb{R}}\ (F)\ = U\Lambda^{+}U^{T}. \tag{2.5}$$

where

$$\Lambda^{+}\ =\ \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.6}$$

and $\Lambda_r\ =\ \text{diag}\ [\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_r]$ is the diagonal matrix formed from the positive eigenvalues of F. Since K_{off} consists of all real symmetric $n \times n$ matrices, in which the off–diagonal elements are fixed to F (the given matrix), therefore

$$P_{\text{off}}\ (A)\ =\ \bar{F}\ +\ \text{Diag}\ A. \tag{2.7}$$

Also, since K_{b} consists of all real symmetric $n \times n$ matrices, in which the diagonal elements are not greater than diag $\mathbf{v} =$ Diag F , we have

$$P_{\text{b}}\ (A)\ =\ \bar{A}\ +\ \text{diag}\ [h_1,\ h_2, ...,\ h_n], \tag{2.8}$$

where

$$\mathbf{h}\ =\ \begin{Bmatrix} h_i & = & a_{ii} & if & a_{ii} & \le & v_i \\ h_i & = & v_i & if & a_{ii} & > & v_i \end{Bmatrix}.$$

3 The l_1SQP Method

This section contains a brief description of the l_1SQP method for solving (1.1).

Problem (1.1) can be expressed as

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{x}^T\mathbf{x} \qquad \mathbf{x}\ \in\ \mathbb{R}^n \\ &\text{subject to}\ \ \bar{A}\ +\ \text{diag}\,\mathbf{x}\ \in\ K_{\mathbb{R}}\ \cap K_{off}(A), \qquad \mathbf{x}\ \le\ \mathbf{v} \end{aligned} \tag{3.1}$$

We can follow [4] for full details in solving (3.1). However, the problems are not the same since the objective function here is quadratic, while it is linear in [4]. Therefore we give a summary of what has been done in [4] with the appropriate changes.

It is difficult to deal with the matrix cone constraints in (3.1) since it is not easy to specify if the elements are feasible or not. Using partial LDL^T factorization of A, this difficulty is rectified. Assume that r, the rank of A^*, is known, then for A sufficiently close to A^*, the partial factors $A = LDL^T$ can be calculated where

$$L = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix}, D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}, A = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix}.$$

Then

$$D_2(A) = A_{22} - A_{21}A_{11}^{-1}A_{21}^T, \tag{3.2}$$

and $D_2(\mathbf{x}) = D_2(\bar{A} + \text{diag } \mathbf{x}) = D_2(A)$. Therefore an equivalent problem to (3.1) with the constraint $D_2 = \mathbf{0}$ is considered and expressed as

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{x}^T\mathbf{x} \qquad \mathbf{x} \in \mathbb{R}^n \\ &\text{subject to} \quad D_2(\mathbf{x}) = 0, \qquad \mathbf{x} \leq \mathbf{v} \end{aligned} \tag{3.3}$$

To eliminate the variables x_i, $i = r+1, \ldots, n$, (3.2) is exploited by using the diagonal elements of $D_2(\mathbf{x})$

$$d_{ii}(\mathbf{x}) = x_i - \sum_{k,l=1}^{r} a_{ik} \, [A_{11}^{-1}]_{kl} \, a_{il} = 0 \qquad i = r+1, \ldots, n \tag{3.4}$$

where a_{ik} and a_{il} are elements in A_{21}. Therefore the unknown variables are reduced to $\mathbf{x} = [x_1, \ x_2, \ \ldots, \ x_r]^T \in \mathbb{R}^r$. This formulation will enable us to derive algorithms with a second order rate of convergence. Now, using the constraint $D_2 = \mathbf{0}$, will produce an equivalent problem to (3.3). The number of variables in this new problem can be reduced to r variables which gives the new reduced problem

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \sum_{k=1}^{r} x_k^2 + \sum_{i=r+1}^{n} x_i^2(\mathbf{x}) \\ &\text{subject to} \quad d_{ij}(\mathbf{x}) = 0, \quad i \neq j, \quad \mathbf{x} \leq \mathbf{v}. \quad i,j = r+1, \ldots, n \end{aligned} \tag{3.5}$$

where $x_i(\mathbf{x})$ indicates that x_i is the function of $\mathbf{x}$ determined by

$$x_i(\mathbf{x}) = \sum_{k,l=1}^{r} a_{ik}[A_{11}^{-1}]_{kl}\ a_{il} \qquad i = r+1,\ldots,n.$$

The Lagrangian for (3.3) is $\mathcal{L}(\mathbf{x},\Lambda,\pi) = \mathbf{x}^T\mathbf{x} - \Lambda : D_2(\mathbf{x}) + \pi^T(\mathbf{x}-\mathbf{v})$. The expressions for the derivatives $\frac{\partial d_{ij}}{\partial x_s}$ and $\frac{\partial^2 d_{ij}}{\partial x_s \partial x_t}$ are given in [4] which enable us to find expressions for $\nabla\ f$, $\nabla^2\ f$ and $W = \nabla^2\mathcal{L}(\mathbf{x},\Lambda,\pi)$, where

$$\nabla\ f \ = \ 2\mathbf{x} \ - \ 2\sum_{i=r+1}^{n} x_i(\mathbf{x})\ \nabla d_{ii}, \tag{3.6}$$

$$\nabla^2\ f \ = \ 2I \ - \ 2\sum_{i=r+1}^{n} [x_i(\mathbf{x})\ \nabla^2 d_{ii} \ - \ (\nabla d_{ii})(\nabla d_{ii})^T] \tag{3.7}$$

and

$$\begin{aligned} W^{(k)} &= \nabla^2\mathcal{L}(\mathbf{x}^{(k)},\Lambda^{(k)},\boldsymbol{\pi}^{(k)}) \\ &= 2I + 2\sum_{i=r+1}^{n} [(\nabla d_{ii}(\mathbf{x}^{(k)}))(\nabla d_{ii}(\mathbf{x}^{(k)}))^T] \ - \sum_{i,j=r+1}^{n} \lambda_{ij}^{(k)} \nabla^2 d_{ij}(\mathbf{x}^{(k)}). \end{aligned} \tag{3.8}$$

Then using these expressions the QP subproblem

$$\begin{aligned} &\underset{\boldsymbol{\delta}}{\text{minimize}} \quad f^{(k)} \ + \ \nabla f^{(k)}\boldsymbol{\delta} \ + \ \tfrac{1}{2}\ \boldsymbol{\delta}^T W^{(k)}\boldsymbol{\delta} \qquad \boldsymbol{\delta} \ \in \ \mathbb{R}^r \\ &\text{subject to} \quad d_{ij}^{(k)} \ + \ \nabla\ d_{ij}^{(k)T}\boldsymbol{\delta} \ = \ 0 \qquad i \ \neq \ j \quad i,j = r+1,\ldots,n \\ &\qquad\qquad \mathbf{x}^{(k)} \ + \ \boldsymbol{\delta} \ \leq \ \mathbf{v} \end{aligned} \tag{3.9}$$

is defined. Thus the SQP method applied to (3.5) requires the solution of the QP subproblem (3.9). The matrix $W^{(k)}$ is positive semi–definite.

4 Hybrid Methods

In this section, a new method for solving (1.1) is introduced. The methods described here depend upon both the projection and l_1SQP methods using a hybrid method.

The hybrid method works in two stages. During the first stage, the projection method converges globally and, hence, is potentially reliable but often converges slowly. During the second stage, the l_1SQP method and the method, described in Section 3, has a second order convergence rate if the correct rank r^* is given. The main disadvantage of the l_1SQP method is that it requires the correct r^*. A hybrid method is one which switches between these methods and aims to combine their best features. To apply the l_1SQP method requires a knowledge of the rank r^* which can be gained from the progress of the projection method.

The main disadvantage of the l_1SQP method is finding the exact rank r^*. Since it is not known in advance, it is necessary to estimate it by an integer $r^{(k)}$. It is suggested that the best estimate of the matrix rank $r^{(k)}$ is obtained by carrying out some iterations of the projection method given in Section 2. This is because the projection method is a globally convergent method.

Considering Λ_r in (2.6), then at the solution, the number of eigenvalues in Λ_r is equal to the rank r^*. Thus

$$No. \quad \Lambda_r^* = r^*, \tag{4.1}$$

where $No.\ \Lambda$ is the number of positive eigenvalues in Λ. An equation similar to (4.1) is used to calculate an estimated rank $r^{(k)}$, given by

$$No. \quad \Lambda_r^{(k)} \; = \; r^{(k)},$$

where Λ_r is given by (2.6). Then, the l_1SQP method will be applied to solve the problem as described in Section 3.

The projection–l_1SQP algorithm can now be described as follows.

Algorithm 4.1 Given any matrix $F = F^T \in \mathbb{R}^{n\times n}$, let s be a positive integer.

Then the following algorithm solves (1.1)

i. Let $F^{(0)} := F$.

ii. Apply Algorithm 2.1 until

$$No. \quad \Lambda_r^{(k)} = No. \ \Lambda_r^{(k+j)} \quad j = 1, 2, \ \ldots, s. \tag{4.2}$$

iii. $r^{(k)} = No. \ \Lambda_r^{(k)}$.

iv. Use the result vector $\mathbf{x}$ from Algorithm 2.1 as an initial vector for the l_1SQP method.

v. Apply the l_1SQP method for solving (1.1).

The integer s in Algorithm 4.1 can be any positive number. If s is small, then the rank $r^{(k)}$ may not be accurately estimated, but the number of iterations taken by projection method is small. On the other hand, if s is large, then a more accurate rank is obtained but the projection method needs more iterations.

The advantage of using the projection method as the first stage of the projection–l_1SQP method is that if $F^{(0)}$ is positive semi–definite and singular of rank r^*, then the projection method terminates at the first iteration. Moreover, it gives the best estimate for $r^{(k)}$.

5 Numerical Results and Comparisons

In this section, numerical problems are obtained from the data given by [8]. The data set is a 64×20 data. Various selections from the set of subsets of columns are used to give various test problems to form the matrix A. These subsets are those given in the first columns of Tables 5.1 and 5.2, the value of n is the number of elements in each subset. Numerical examples for Algorithm 4.1 are given in some detail in Table 5.1. Also the same numerical examples are given in Table 5.2. for Algorithm 2.1, l_1SQP algorithm and Algorithm 4.1.

The results obtained by Algorithm 4.1 are tabulated in Table 5.1. Using $\|\mathbf{x}^{(k+1)}\| - \|\mathbf{x}^{(k)}\| < 10^{-8}$ as a stopping criterion it is estimated that the x_i are accurate to $4-5$ decimal places and $\|\mathbf{x}\|_2$ is accurate to $6-7$ decimal places. In Table 5.1 the column headed by NI gives the number of iterations used by the projection method. It is clear from Table 5.1 that when the bounds are active the number of iterations becomes very large. The x_i^* elements marked by $(*)$ are the active elements.

Moreover Table 5.1 gives the correct rank r^* for each particular problem. The order of convergence is very slow as seen in Table 5.1. Also in Table 5.1 the optimal x_i^* for $i = 1, 2, ..., n$ and $\|\mathbf{x}^*\|_2$ are given. The eigenvalues for the projection method are solved using the NAG library.

In Table 5.2 three methods are compared: Algorithm 2.1 (PM), l_1SQP algorithm and Algorithm 4.1 (Pl_1SQP). In Table 5.2 the columns headed by NI give the number of iterations used by the projection method and the columns headed by NQP gives the number of times that the major l_1SQP is solved. $r^{(0)}$ in the column headed by l_1SQP gives the initial rank for F. $r^{(0)}$ in the column headed by Pl_1SQP gives the initial rank for F using Algorithm 4.1. The three methods converge to approximately the same values.

In l_1SQP one of the variables in almost every test example is adjusted by a small unit (< 2.0) so that the matrix $\bar{A} + \text{diag } \mathbf{x}^*$ is exactly singular and positive semi–definite for all methods. In l_1SQP most cases require a few iterations for solving (3.5) as r increases. For each value of r second order convergence is obtained.

Columns which determine F	r^*	NI	x_i^* $i = 1, 2, .., n$				$\sqrt{\sum (x_i^{2*})}$
1,2,5,6	3	63	182.7042	146.9628	69.6629	45.8211	248.8602
1,3,4,5	2	115	235.0096	88.4015	189.1918	67.6986	321.5913
1,2,3,6,8,10	5	141	367.4156	273.0114	279.8192	50.4784	616.2334
			228.0582	193.2790			
1,2,4,5,6,8	4	881	317.4348	146.2721	244.8117	65.6893	491.7348
			4.1061	235.3253			
1–6	5	336	222.2243	282.8910	262.8245	238.0719	510.3758
			71.5195	14.2313			
1–8	6	387	369.8391	290.2214	255.5179	176.0771	640.5922
			56.6419	48.0679	223.0925	194.3380	
1–10	8	954	401.7844	299.7303	249.6374	194.1057	736.9839
			35.6192	50.3791	240.8572	214.9912	
			232.9831	171.9279			
1–12	10	1360	386.8981	286.8628	264.6721	195.7548	800.0756
			67.2526	39.7566	232.4680	227.8524	
			266.8375	187.5834	131.9821	252.7745	
1–14	12	854	404.4696	294.5210	265.8667	213.4180	882.7606
			73.4999	35.6596	254.5520	235.9188	
			250.0652	191.7257	161.8923	250.0233	
			267.8237	160.7042			
1–16	14	3663	407.5394(*)	290.8398	275.5972	215.0889	945.4555
			81.3601	33.5239	248.6281	244.9842	
			261.4713	197.1172	168.2075	258.6026	
			259.0489	159.3373	99.1123	294.4601	
1–18	15	30326	407.5394(*)	296.5150	265.6089	216.2863	1108.5326
			98.2078	44.7847	260.8753	246.8023	
			248.7318	185.1102	176.9004	270.7481	
			258.8518	160.6789	101.7151	308.4449	
			435.4937	358.0457			
1–20	18	11037	407.5394(*)	312.4666	258.1156	227.1807	1253.6603
			120.1546	49.2651	292.7023	272.3617	
			244.4578	201.3850	175.7458	279.3872	
			250.5748	158.5493	100.0581	310.8974	
			457.7386	356.8083	406.2569	327.4915	

Table 5.1: Results for (1.1) from projection Algorithm 2.1.

Columns which determine A	r^*	PM NI	l_1SQP $r^{(0)}$	l_1SQP NQP	Pl_1SQP NI	Pl_1SQP $r^{(0)}$	Pl_1SQP NQP
1,2,5,6	3	63	2	10	5	3	4
1,3,4,5	2	115	2	16	6	2	5
1,2,3,6,8,10	5	141	3	11	10	4	9
1,2,4,5,6,8	4	881	3	20	8	4	7
1–6	5	336	3	22	12	5	9
1–8	6	387	5	18	13	5	11
1–10	8	954	6	19	7	8	7
1–12	10	1360	8	27	16	8	24
1–14	12	854	10	30	20	10	14
1–16	14	3663	11	35	27	10	33
1–18	15	30326	13	33	38	12	13
1–20	18	11037	15	45	55	15	27

Table 5.2: Numerical comparisons between methods of this paper.

The projection method is a very slowly convergent method especially when the bounds are active. Therefore it will be used only for estimating the rank r.

6 Conclusions

In this paper we have studied certain problems involving the positive semi–definite matrix constraint. One is the projection method, and the other is l_1SQP method. The hybrid method developed in Section 4 give a good rate of convergence as compared with the methods of Sections 2 and 3. The projection method is not very effective in determining the rank when $n \geq 12$ and a more effective method is required to give a better estimate for r^*.

Acknowledgements The author is grateful to King Fahd University of Petroleum and Minerals for providing research facilities.

References

[1] Al-Homidan, S. [1998]. Hybrid methods for Solving the Educational Testing Problem, *J. Comp. App. Math.,* To appear.

[2] Al-Homidan, S. and Fletcher, R. [1995]. Hybrid methods for finding the nearest Euclidean distance matrix, in *Recent Advances in Nonsmooth Optimization* (Eds. D. Du, L. Qi and R. Womersley), World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 1–17.

[3] Dykstra, R. L. [1983]. An algorithm for restricted least squares regression, *J. Amer. Stat. Assoc. 78, pp. 839–842.*

[4] Fletcher, R. [1985]. Semi–definite matrix constraints in optimization, *SIAM J. Control and Optimization, 23, pp. 493–513.*

[5] Han, S. P. A successive projection method, *Math. Programming, 40, pp. 1–14.*

[6] Higham, N. [1988]. Computing a nearest symmetric positive semi–definite matrix, *Linear Alg. and Appl., 103, pp. 103–118.*

[7] Von Neumann, J [1950]. *Functional Operators II. The Geometry of Orthogonal Spaces,* Annals of Math. Studies No. 22, Princeton Univ. Press.

[8] Woodhouse, B. [1976]. Lower bounds for the reliability of a test, M.Sc. Thesis, Dept of Statistics, University of Wales, Aberystwyth.