Chapter 5

Methods for minimizing least distance functions with semi–definite matrix constraints

5.1 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. We are interested here in problems in which only the diagonal of the matrix is allowed to change, in the following way. Given a symmetric positive definite matrix $F \in \Re^{n \times n}$ then we consider the problem

$$\begin{aligned} &\textit{minimize} \quad f(\mathbf{x}) \\ &\textit{subject to} \;\; \bar{F} + \textit{diag}\,\mathbf{x} \geq 0 \\ &\qquad\qquad x_i \leq v_i \qquad i = 1, ..., n \end{aligned} \tag{5.1.1}$$

where $\bar{F} = F - \textit{Diag}\ F$, $\textit{diag}\,\mathbf{v} = \textit{Diag}\ F$ and f is real valued function of $\mathbf{x}$. In Chapters 6 and 7, such problems are studied in which the objective function is linear.

In this chapter a least distance problem of the following type is solved. Given a symmetric positive semi–definite matrix $F \in \Re^{n \times n}$ then we consider

$$\begin{aligned} &minimize \quad \mathbf{x}^T\mathbf{x} \qquad \mathbf{x} \in \Re^n \\ &subject\ \ to \ \ \bar{F} + diag\,\mathbf{x} \geq 0 \\ &\qquad\qquad x_i \leq v_i \qquad i = 1, ..., n \end{aligned} \tag{5.1.2}$$

where $diag\ \mathbf{v} = Diag\ F.$ This kind of problem is important by itself and it is also used subsequently in Chapters 6 and 7. Problem (5.1.2) can be more general if we express it as

$$\begin{aligned} &minimize \quad \|\mathbf{a} - \mathbf{x}\|_2^2 \qquad \mathbf{x} \in \Re^n \\ &subject\ \ to \ \ \bar{F} + diag\,\mathbf{x} \geq 0 \\ &\qquad\qquad x_i \leq v_i \qquad i = 1, ..., n \end{aligned} \tag{5.1.3}$$

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 chapter.

Two methods are developed for solving problem (5.1.2). Firstly, a projection algorithm is given for solving problem (5.1.2) using Algorithm 2.2.7 which converges linearly or slower and globally. This method is described in Section 5.2. Subsequently this method is also used in Chapter 6. Secondly an implementation of the l_1 SQP method is used. Fletcher [1985] developed an algorithm for solving problem (5.1.1) in the case $f(\mathbf{x})$ is linear. It is the purpose of this chapter to follow his method but to apply it to problem (5.1.2). Various methods of this type are investigated in Section 5.3.

In Section 5.4 a hybrid method is described, which starts with the projection method to estimate the rank $r^{(k)}$ and continues with the l_1SQP method in a similar way to Section 4.3. Finally in Section 5.5 numerical comparisons of these methods are carried out.

5.2 The Projection algorithm

In this section we give a description of a projection algorithm for solving problem (5.1.2), using the alternating projection method of Algorithm 2.2.7. The constraints in problem (5.1.2) can be expressed as $\bar{F} + diag\ \mathbf{x} \in K_{\Re} \cap K_{off} \cap K_b$ which gives an equivalent problem to (5.1.2) and can be expressed as

Given a symmetric positive definite matrix $F = F^T \in \Re^{n \times n}$

$$\begin{aligned} &minimize \quad \|\bar{F} - A\| \\ &subject \ \ to \ \ A \in K_{\Re} \cap K_{off} \cap K_b. \end{aligned} \tag{5.2.1}$$

The matrix norm here means the Frobenius norm given in Definition 1.2.2.

Then we follow Algorithm 2.2.7 with $m = 3$ and $K_1 = K_{\Re}$, $K_2 = K_{off}$ and $K_3 = K_b$ as given in (1.3.1), (1.3.5) and (1.3.6) respectively. Algorithm 2.2.7 is the projection algorithm used in this section, and guarantees global convergence to the solution of problem (5.1.2). The projection algorithm requires formulae, which are also given, for calculating the projection maps on to K_{off}, K_b and on to $K_{\Re}$. Subsequently two examples are given for solving problem (5.1.2) using the projection algorithm. Finally an interesting result relating normal cone of the intersection of K_{off}, K_b and $K_{\Re}$ to the solution of problem (5.2.1) is given.

Dykstra's algorithm depends crucially upon the computational complexity of the relevant projections. The minimization problem (5.2.1) is solved by applying Algorithm 2.2.7 to it. Problem (5.2.1) is to find the projection of a matrix to the intersection of three convex sets by a sequence of projections to the individual set successively. First we need definitions for the projection maps $P_{\Re}(\cdot)$, $P_{off}(\cdot)$ and $P_b(\cdot)$, later formulae for them are obtained.

Definition 5.2.1

Let

$$K = \{A : A \in \Re^{n \times n}, A = A^T\},$$

then define the projection map $P_{\Re}(A)$ from K on to $K_{\Re}$, the projection map $P_{off}(A)$ from K on to K_{off} and the projection map $P_b(A)$ from K on to K_b.

The projection map $P_{\Re}(A)$ formula on to $K_{\Re}$ for solving the following problem

$$\begin{aligned} &minimize \ \|F - A\|_F \\ &subject \ \ to \ \ A \in K_{\Re} \end{aligned} \tag{5.2.2}$$

is

$$P_{\Re}\ (F)\ = U\Lambda^{+}U^{T}. \tag{5.2.3}$$

where

$$\Lambda^{+}\ =\ \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.2.4}$$

and $\Lambda_r\ =\ diag\ [\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_r]$ is the diagonal matrix formed from the positive eigenvalues of F. The proof has been given in Theorem 3.3.3.

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) then

$$P_{off}\ (A)\ =\ \bar{F}\ +\ Diag\ A. \tag{5.2.5}$$

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

$$P_b\ (A)\ =\ \bar{A}\ +\ diag\ [h_1,\ h_2, ...,\ h_n]. \tag{5.2.6}$$

where

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

We can now use projections $P_{\Re},\ P_{off}$ and P_b given by (5.2.3), (5.2.5) and (5.2.6) respectively to implement Algorithm 2.2.7 giving the following algorithm

Algorithm 5.2.2 (*projection algorithm*)

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

```
$For\ \ k\ =\ 0,\ 1,\ 2,\ \ \ldots$
$F^{(k+1)}\ =\ F^{(k)}\ +\ [P_b P_{off} P_{\Re}(F^{(k)})\ -\ P_{\Re}(F^{(k)})]$
```

The convergence of this algorithm follows from Theorem 2.2.8 in which the sequences $\{P_{\Re}(F^{(k)})\}, \{P_{off}\ P_{\Re}(F^{(k)})\}$ and $\{P_b\ P_{off}\ P_{\Re}(F^{(k)})\}$ generated by Algorithm 5.2.2 converge in the Frobenius norm to the solution A^* of (5.2.1).

Example 5.2.3

An example of Algorithm 5.2.2 for $n = 3$, let

$$\bar{F} = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}.$$

The solution is $\mathbf{x}^* = (3,\ 4/3,\ 3)$, no bounds are active, the rank of $F^* = \bar{F} + diag\,\mathbf{x}^*$ is $r = 1$.

Example 5.2.4

Another example for $n = 4$, let

$$\bar{F} = \begin{bmatrix} 0 & 1 & 2 & -2 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ -2 & 2 & 1 & 0 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} 2 \\ 4 \\ 8 \\ 10 \end{bmatrix}.$$

The solution is $\mathbf{x}^* = (2,\ 2.6505,\ 4.1209,\ 6.3537)$. The bound $x_1 \leq v_1$ is active. If v_1 is increased to $v_1 = 5$ then the bound $x_1 \leq v_1$ is not active and the new solution for this modified problem is

$$\mathbf{x}^* = (3.4555,\ 3.1833,\ 3.1833,\ 3.4555).$$

The rank of $F^* = \bar{F} + diag\,\mathbf{x}^*$ is $r = 2$ in both cases.

In the rest of this section another result is developed giving conditions under which A^* solves (5.2.1). The normal cone $\partial K_{\Re}(A)$ at $A \in K_{\Re}$ is given in (1.3.12). Also the normal cone for $K_{off} \cap K_b$ is given in Theorem 1.3.5. This is based on normal cones for the relevant convex sets. A general result for the normal cone of the intersection of two sets has been given in (1.3.9). Therefore, as in Theorem 3.3.1, if $A \in K_{\Re} \cap K_{off} \cap K_b$ then

$$\partial(K_{\Re} \cap K_{off} \cap K_b)(A) = \partial K_{\Re}(A) + \partial(K_{off} \cap K_b)(A) \qquad (5.2.7)$$

Now $\partial K_{\Re}(A)$ and $\partial(K_{off} \cap K_b(A))(A)$ are given in (1.3.12) and (1.3.16) respectively and we let Z, Λ and B denote the matrices that arise. From (5.2.7) and (2.1.3) we can deduce that A^* solves problem (5.2.1) if and only if

$$F - A^* \ = -Z \, \Lambda \, Z \ + \ B \ = \ U \ \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\Lambda \end{bmatrix} U^T \ + \ B \tag{5.2.8}$$

where $U \ = \ [Y \quad Z]$ as in (1.3.15). Then (5.2.8) is equivalently to

$$F \ = \ U \ \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & -\Lambda \end{bmatrix} U^T \ + \ B$$

since

$$A^* \ = \ U \ \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} U^T, \tag{5.2.9}$$

from the spectral decomposition of A^* since $A^* \ \in \ K_{\Re}$ from (5.2.1) and U is the same as U in (5.2.8) from Theorem 1.3.7.

5.3 The l_1SQP method

The main idea in this section is to find an algorithm which is globally convergent at a second order rate for solving problem (5.1.2). The idea of transforming the semi–definite matrix constraints in to the form $D_2(A) \ = \ \mathbf{0}$ given in (1.5.10) is used. The SQP methods in Section 1.7 are used in order to have the benefit of the ready availability of second derivatives of (1.5.10) which enables a second order rate of convergence to be achieved. At the end of this section a strategy is described of how to choose the rank r needed to determine D_2. Also two examples for solving problem (5.1.2) are given which are similar to Examples 5.2.3 and 5.2.4. However in the first part of this section we consider the normal cone and the feasible directions sets for the special case in which the positive semi–definite matrix cone $K_{\Re}$ is restricted to the diagonal elements of A (i. e. $A \ \in \ K_{\Re} \ \cap \ K_{off}$).

Now problem (5.2.1) can be expressed as

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{x}^T\mathbf{x} \qquad \mathbf{x} \ \in \ \Re^n \\ &\textit{subject to} \ \ \bar{A} \ + \ \textit{diag}\, \mathbf{x} \ \in \ K_{\Re} \cap K_{off}(A), \qquad \mathbf{x} \ \leq \ \mathbf{v} \end{aligned} \tag{5.3.1}$$

where $diag\, \mathbf{v} \; = \; Diag\; F$ (the given matrix).

A useful form of $\partial(K_{\Re} \cap K_{off})(A)$ can be deduced using (1.3.12), let $\bar{B} \; = \; B \; - \; Diag\; B$ then

$$\partial(K_{\Re} \cap K_{off})(A) \; = \; \{\acute{B}|\; Diag\; \acute{B} \; = \; B \; - \; \bar{B}, \;\; B \; = \; -Z\Lambda Z^T, \;\; \Lambda \; = \; \Lambda^T, \; \Lambda \; \geq \; 0\} \tag{5.3.2}$$

that is the set of the vectors that are diagonal elements of all matrices of the form $-Z\Lambda Z^T$, where Λ is any symmetric positive semi–definite matrix and Z is the null space matrix.

Furthermore feasible directions for the set $K_{\Re} \cap K_{off}(A)$ can be deduced using (1.4.5)

$$\mathcal{F}(A) \; = \; F(A) \; = \; \{\bar{A} \; + \; diag\; \mathbf{s}|\; Z^T[diag\; \mathbf{s}]Z \; \geq \; 0\}. \tag{5.3.3}$$

Optimality conditions follow using Theorem 1.5.2. The first order necessary conditions for $\mathbf{x}^*$ to solve (5.3.1) are that $\mathbf{x}^*$ is feasible and there exist a matrix $\acute{B}^* \; \in \; \partial(K_{\Re} \cap K_{off})(A^*)$ and a vector $\boldsymbol{\pi}^* \; \geq \; 0$ $(\boldsymbol{\pi}^* \; \in \; \Re^n)$ such that

$$2\mathbf{x}^* \; + \; \mathbf{b}^* \; + \; \boldsymbol{\pi}^* \; = \; 0 \tag{5.3.4a}$$

$$\boldsymbol{\pi}^{*T} \; (\mathbf{v} - \mathbf{x}^*) \; = \; 0 \tag{5.3.4b}$$

where $diag\; \mathbf{b}^* \; = \; Diag\; \acute{B}^*$.

Now we going to use the second derivatives of (1.5.10) to solve problem (5.3.1).

Assume that the rank of A^* is known to be r $(1 \; \leq \; r \; < n)$. Permute the variables so that the bounds $x_i \; \leq \; v_i$ are inactive for $i = r+1, \ldots, \; n,$ then (5.3.1) can be expressed as

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{x}^T\mathbf{x} & \mathbf{x} \; \in \; \Re^n \\ subject \;\; to \;\; D_2(\mathbf{x}) \; = \; 0, & \mathbf{x} \; \leq \; \mathbf{v} \end{array} \tag{5.3.5}$$

where

$$D_2(\mathbf{x}) \; = \; D_2(\bar{A} \; + \; diag\; \mathbf{x}) \; = \; D_2(A)$$

and $D_2(A)$ is given by (1.5.9). The Lagrangian for problem (5.3.5) is

$$\mathcal{L}(\mathbf{x}, \Lambda, \boldsymbol{\pi}) \; = \; \mathbf{x}^T\mathbf{x} \; - \; < \Lambda, D_2(\mathbf{x}) > \; + \; \boldsymbol{\pi}^T(\mathbf{x} \; - \; \mathbf{v}). \tag{5.3.6}$$

Also, the first order conditions for this problem are given by (5.3.4a) and (5.3.4b). From (5.3.2) $Diag\ \acute{B}$ is a diagonal matrix which has the same elements as the diagonal of the matrix $-Z^T \Lambda^* Z$ where $\Lambda^*(= [\lambda^*_{ij}] \quad i, j = r+1, \ldots, n)$ is the matrix of Lagrange multipliers for the constraints $D_2(\mathbf{x}) = 0$ and Z is the null space matrix for A^*. The elements of the Lagrange matrix Λ are indexed from $r+1, \ldots, n$ to correspond to the elements d_{ij} of D_2. Then using (1.5.9) in (5.3.6)

$$\frac{\partial \mathcal{L}}{\partial x_i} = 2x_i - \lambda_{ii} + \pi_i = 0. \qquad i = r+1, \ldots, n \tag{5.3.7}$$

The assumption that the bounds are inactive at the solution for $i > r$ i.e. $\pi_i = 0$ implies that

$$\lambda_{ii} = 2x_i. \qquad i = r+1, \ldots, n \tag{5.3.8}$$

To eliminate the variables $x_i, \ i = r+1, \ldots, n$ (1.5.9) is utilized by using the diagonal elements of $D_2(\mathbf{x})$

$$d_{ii}(\mathbf{x}) = x_i - \sum_{k,l=1}^{r} a_{ik} \, [A_{11}^{-1}]_{kl} \ a_{il} = 0 \qquad i = r+1, \ldots, n \tag{5.3.9}$$

where a_{ik} *and* a_{il} are elements in A_{21}. Therefore the unknown variables are reduced to $\mathbf{x} = [x_1, \ x_2, \ \ldots, \ x_r]^T \in \Re^r$. Then (5.3.5) reduces to

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \sum_{k=1}^{r} x_k^2 + \sum_{i=r+1}^{n} x_i^2(\mathbf{x}) \\ &\textit{subject to } d_{ij}(\mathbf{x}) = 0, \quad i \neq j, \quad i, j = r+1, \ldots, n \\ &\qquad\qquad \mathbf{x} \leq \mathbf{v} \end{aligned} \tag{5.3.10}$$

the alternative unknown vector is determined by (5.3.9). $x_i(\mathbf{x})$ denotes that x_i is the function of $\mathbf{x}$ given by

$$x_i(\mathbf{x}) = \sum_{k,l=1}^{r} a_{ik} \, [A_{11}^{-1}]_{kl} \ a_{il} \qquad i = r+1, \ldots, n \tag{5.3.11}$$

where $Diag\ A_{11} \ = \ diag\ \mathbf{x}$.

In (5.3.10) the constraints $d_{ij}(\mathbf{x}) \ = \ 0$ and $d_{ji}(\mathbf{x}) \ = \ 0$ are both equivalent, therefore in practice the constraints should be presented only for $i \ > \ j$ with $2\lambda_{ij}$ as the Lagrange multiplier for each constraint in this system. However in the rest of this section it is more convenient to refer to (5.3.10).

If

$$\Lambda \ = \ \begin{bmatrix} 2x_{r+1}(\mathbf{x}) & \dots & \dots & \lambda_{r+1\ n} \\ \vdots & \ddots & \vdots & \vdots \\ \lambda_{n-1\ r+1} & \dots & \dots & \lambda_{n+1\ n} \\ \lambda_{n\ r+1} & \dots & \lambda_{n\ n-1} & 2x_n(\mathbf{x}) \end{bmatrix}$$

then (5.3.6) is the Lagrangian function for (5.3.10).

In the following expressions for ∇d_{ij} and $\nabla^2 d_{ij}$ will be derived where ∇ denotes the gradient operator $(\partial/\partial x_1, \ \dots, \ \partial/\partial x_r)^T$. Differentiating $A_{11}A_{11}^{-1} \ = \ I$ gives

$$\frac{\partial A_{11}}{\partial x_s} A_{11}^{-1} \ + \ A_{11}\frac{\partial A_{11}^{-1}}{\partial x_s} \ = \ 0 \qquad\qquad s \ = \ 1, \dots, r$$

$$\Rightarrow \qquad\qquad A_{11}\frac{\partial A_{11}^{-1}}{\partial x_s} \ = \ - \ \frac{\partial A_{11}}{\partial x_s} A_{11}^{-1}$$

then

$$\frac{\partial A_{11}^{-1}}{\partial x_s} \ = \ - \ A_{11}^{-1}\ \frac{\partial A_{11}}{\partial x_s} A_{11}^{-1},$$

but since

$$\frac{\partial A_{11}}{\partial x_s} \ = \ \mathbf{e}_s\mathbf{e}_s^T$$

where $\mathbf{e}_s \ = \ (0, \ 0, \ \dots, \ 0, \ 1, \ 0, \ \dots, \ 0)$ with one in the sth component, then

$$\frac{\partial A_{11}^{-1}}{\partial x_s} \ = \ - \ A_{11}^{-1}\ \mathbf{e}_s\mathbf{e}_s^T\ A_{11}^{-1}. \tag{5.3.12}$$

Hence from (1.5.9)

$$\begin{aligned}\frac{\partial D_2}{\partial x_s} &= \frac{\partial}{\partial x_s}(A_{22} \ - \ A_{21}A_{11}^{-1}A_{21}^T) \\ &= 0 \ - \ A_{21}\ \frac{\partial A_{11}^{-1}}{\partial x_s}\ A_{21}^T \\ &= A_{21}A_{11}^{-1}\ \mathbf{e}_s\mathbf{e}_s^T\ A_{11}^{-1}A_{21}^T\end{aligned}$$

Using (1.5.11) gives

$$\frac{\partial D_2}{\partial x_s} = V_{21}^T\ \mathbf{e}_s\mathbf{e}_s^T\ V_{21}$$

and hence

$$\frac{\partial d_{ij}}{\partial x_s} = v_{si}\ v_{sj}. \tag{5.3.13}$$

Furthermore differentiating (5.3.12)

$$\begin{aligned}\frac{\partial^2 A_{11}^{-1}}{\partial x_s \partial x_t} &= \frac{\partial}{\partial x_t}(-\ A_{11}^{-1}\ \mathbf{e}_s\mathbf{e}_s^T\ A_{11}^{-1}) \\ &= -[(-A_{11}^{-1}\ \mathbf{e}_t\mathbf{e}_t^T\ A_{11}^{-1})\ \mathbf{e}_s\mathbf{e}_s^T\ A_{11}^{-1} \ + \ A_{11}^{-1}\ \mathbf{e}_s\mathbf{e}_s^T\ (-A_{11}^{-1}\ \mathbf{e}_t\mathbf{e}_t^T\ A_{11}^{-1})] \\ &= A_{11}^{-1}(\mathbf{e}_t\mathbf{e}_t^T\ A_{11}^{-1}\ \mathbf{e}_s\mathbf{e}_s^T \ + \ \mathbf{e}_s\mathbf{e}_s^T\ A_{11}^{-1}\ \mathbf{e}_t\mathbf{e}_t^T)A_{11}^{-1}.\end{aligned}$$

So from (1.5.9)

$$\begin{aligned}\frac{\partial^2 D_2}{\partial x_s \partial x_t} &= -\ A_{21}A_{11}^{-1}(\mathbf{e}_t\mathbf{e}_t^T\ A_{11}^{-1}\ \mathbf{e}_s\mathbf{e}_s^T \ + \ \mathbf{e}_s\mathbf{e}_s^T\ A_{11}^{-1}\ \mathbf{e}_t\mathbf{e}_t^T)A_{11}^{-1}A_{21}^T \\ &= -\ V_{21}^T(\mathbf{e}_t\mathbf{e}_t^T\ A_{11}^{-1}\ \mathbf{e}_s\mathbf{e}_s^T \ + \ \mathbf{e}_s\mathbf{e}_s^T\ A_{11}^{-1}\ \mathbf{e}_t\mathbf{e}_t^T)V_{21}\end{aligned}$$

hence

$$\frac{\partial^2 d_{ij}}{\partial x_s \partial x_t} = -\ (v_{si}\ v_{tj} \ + \ v_{ti}\ v_{sj})[A_{11}^{-1}]_{st}. \tag{5.3.14}$$

where $[A_{11}^{-1}]_{st}$ means the element of A_{11}^{-1} in st position.

For the SQP method the solution of the QP subproblem (1.7.8) is needed. In (1.7.8) $c_i = d_{ij}$ and $\nabla c_i = \mathbf{a}_i^T = \nabla d_{ij}^T$, which are given in (5.3.13). From (5.3.9), (5.3.10) and (5.3.11)

$$\nabla\ f \ = \ 2\mathbf{x} \ + \ 2 \sum_{i=r+1}^{n} x_i(\mathbf{x}) \ \nabla x_i(\mathbf{x})$$

then

$$\nabla\ f \ = \ 2\mathbf{x} \ - \ 2 \sum_{i=r+1}^{n} x_i(\mathbf{x}) \ \nabla d_{ii} \tag{5.3.15}$$

and

$$\nabla^2\ f \ = \ 2I \ - \ 2 \sum_{i=r+1}^{n} [x_i(\mathbf{x}) \ \nabla^2 d_{ii} \ - \ (\nabla d_{ii})(\nabla d_{ii})^T] \tag{5.3.16}$$

Now in the QP subproblem (1.7.8) $W = \nabla^2 \mathcal{L}(\mathbf{x}, \Lambda, \boldsymbol{\pi})$ then from (5.3.6) and (5.3.16)

$$W^{(k)} \ = \ \nabla^2 \mathcal{L}(\mathbf{x}^{(k)}, \Lambda^{(k)}, \boldsymbol{\pi}^{(k)})$$

$$= \ 2I \ - \ 2 \sum_{i=r+1}^{n} [x_i(\mathbf{x}^{(k)}) \ \nabla^2 d_{ii}(\mathbf{x}^{(k)}) \tag{5.3.17}$$

$$- \ (\nabla d_{ii}(\mathbf{x}^{(k)}))(\nabla d_{ii}(\mathbf{x}^{(k)}))^T] \ - \sum_{\substack{i,j=r+1 \\ i\neq j}}^{n} \lambda_{ij}^{(k)} \nabla^2 d_{ij}(\mathbf{x}^{(k)}). \tag{5.3.18}$$

Including term (5.3.17) in the diagonal of the last term of (5.3.18) with $\lambda_{ii}^{(k)} = 2x_i(\mathbf{x}^{(k)})$ (from (5.3.8)) gives

$$W^{(k)} \ = \ 2I \ + \ 2 \sum_{i=r+1}^{n} [(\nabla d_{ii}(\mathbf{x}^{(k)}))(\nabla d_{ii}(\mathbf{x}^{(k)}))^T] \ - \sum_{i,j=r+1}^{n} \lambda_{ij}^{(k)} \nabla^2 d_{ij}(\mathbf{x}^{(k)}). \tag{5.3.19}$$

Now

$$\sum_{i=r+1}^{n} [(\nabla d_{ii}(\mathbf{x}^{(k)}))(\nabla d_{ii}(\mathbf{x}^{(k)}))^T \ = \ \begin{bmatrix} \sum_i v_{1i}^2 v_{1i}^2 & \dots & \sum_i v_{1i}^2 v_{ri}^2 \\ \vdots & \ddots & \vdots \\ \sum_i v_{ri}^2 v_{1i}^2 & \dots & \sum_i v_{ri}^2 v_{ri}^2 \end{bmatrix}$$

$$= \ UU^T \tag{5.3.20}$$

since $\partial d_{ii}/\partial x_s = v_{si}^2$, where $U = [V_{12}][V_{12}]$ and $[\ ][\ ]$ means the componentwise product. Rearranging (5.3.19) using (5.3.20) and (5.3.14) gives

$$\begin{aligned}
[W^{(k)}]_{st} &= [2I]_{st} + 2[UU^T]_{st} + 2[V_{12}\Lambda^{(k)}V_{12}^T]_{st}[A_{11}^{-1}]_{st} \\
&= [2I]_{st} + 2[UU^T]_{st} + 2[V_{12}\Lambda^{(k)}V_{12}^T]_{st}[V_{11}D_1^{-1}V_{11}^T]_{st} \qquad (5.3.21)
\end{aligned}$$

where $s,t = 1,\ldots,r$. V and D in (5.3.21) are calculated using (1.5.11) and (1.5.4–5).

From the above expressions the QP subproblem (1.7.8) can be expressed as

$$\begin{aligned}
&\underset{\boldsymbol{\delta}}{\text{minimize}} \quad f^{(k)} + \nabla f^{(k)}\boldsymbol{\delta} + \tfrac{1}{2}\,\boldsymbol{\delta}^T W^{(k)}\boldsymbol{\delta} \qquad\qquad \boldsymbol{\delta} \in \Re^r \\
&\textit{subject to} \;\; d_{ij}^{(k)} + \nabla\, d_{ij}^{(k)T}\boldsymbol{\delta} = 0 \qquad i \neq j \quad i,j = r+1,\ldots,n \\
&\qquad\qquad\qquad \mathbf{x}^{(k)} + \boldsymbol{\delta} \leq \mathbf{v} \qquad\qquad\qquad\qquad\qquad (5.3.22)
\end{aligned}$$

giving a correction vector $\boldsymbol{\delta}^{(k)}$, so that $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\delta}^{(k)}$. Further the Lagrange multipliers of the equations in (5.3.22) become the elements $\lambda_{ij}^{(k+1)}$ for the next iteration.

The matrix W^* is positive semi–definite. This can be proved using (5.3.21) because

$$\mathbf{z}^T W^* \mathbf{z} = 2\mathbf{z}^T\mathbf{z} + 2\mathbf{z}^T UU^T\mathbf{z} + 2\mathbf{z}^T[V_{12}\Lambda^* V_{12}^T][V_{11}D_1^{-1}V_{11}^T]\mathbf{z}. \qquad (5.3.23)$$

Since $\mathbf{z}^T UU^T\mathbf{z} \geq 0$ and from (5.3.2) $\Lambda^* \geq 0$ then

$$\begin{aligned}
\mathbf{z}^T W^* \mathbf{z} &= 2\mathbf{z}^T\mathbf{z} + 2\mathbf{z}^T UU^T\mathbf{z} + 2\, tr(V_{12}\Lambda^* V_{12}^T[diag\ \mathbf{z}]V_{11}D_1^{-1}V_{11}^T[diag\ \mathbf{z}]) \\
&= 2\mathbf{z}^T\mathbf{z} + 2\mathbf{z}^T UU^T\mathbf{z} \\
&\qquad + 2\, tr(D_1^{-1/2}V_{11}^T[diag\ \mathbf{z}]V_{12}\Lambda^* V_{12}^T[diag\ \mathbf{z}]V_{11}D_1^{-1/2}) \\
&\geq 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (5.3.24)
\end{aligned}$$

since

$$\{D_1^{-1/2}V_{11}^T[diag\ \mathbf{z}]V_{12}\}\ \Lambda^*\ \{V_{12}^T[diag\ \mathbf{z}]V_{11}D_1^{-1/2}\}^T$$

is symmetric and positive semi–definite. Therefore if $\mathbf{x}^{(k)}$ is sufficiently close to $\mathbf{x}^*$ the basic SQP method converges and the rate is second order (see Section 1.7).

It is shown in Section 1.7 that the SQP method may not converge globally and it is usually modified by the l_1 exact penalty function. An equivalent form to (1.7.11) for problem (5.3.10) is

$$\begin{aligned}\phi(\mathbf{x}) \;=\; & \sum_{k=1}^{r} x_k^2 \;+\; \sum_{i=r+1}^{n} x_i^2(\mathbf{x}) \\ & +\; \sigma\{ \sum_{\substack{i,j=r+1\\ i\neq j}}^{n} |d_{ij}(\mathbf{x})| \;+\; \sum_{i=r+1}^{n} \max(v_i \;-\; x_i,\; 0)\}. \qquad (5.3.25)\end{aligned}$$

Since the bounds are inactive for $i \;>\; r,\;\; \pi_i$ is zero, implying that the max terms are zero if $\mathbf{x}^{(k)}$ is sufficiently close to $\mathbf{x}^*$. To guarantee that the minimizer $\mathbf{x}^*$ of (5.3.25) satisfies first order conditions for (5.3.10), the penalty parameter σ in (5.3.25) must satisfy

$$\sigma \;\geq\; \max_{ij} |\lambda^*_{ij}|. \qquad i,j \;=\; r+1,\ldots,n$$

Now since $\Lambda^* \;\geq\; 0$ and $\lambda^*_{ii} \;=\; 2x^*_i \quad i \;=\; r+1,\; \ldots,\; n$ then

$$\max_{ij} |\lambda^*_{ij}| \;\leq\; 2\; \max_{i}\; x^*_i. \qquad i,j \;=\; r+1,\ldots,n$$

Hence $\sigma \;\geq\; 2\; \max_i\; x^*_i$ must hold. However, since it is advantageous to choose σ as small as possible, the choice $\sigma \;=\; 2\; \max_i\; x^*_i$ is recommended. In practice if the unnecessarily redundant form of (5.3.10) is used with summation over indices $i \;>\; j$, then a similar summation is used in (5.3.25) and the choice $\sigma \;=\; 4\; \max_i\; x^*_i$ is recommended.

To ensure the descent property, it may be necessary to choose larger values of σ than $\sigma \;=\; 4\; \max_i\; x^*_i$ the choice

$$\sigma \;>\; \max_{ij} |\lambda^{(k+1)}_{ij}| \qquad i,j \;=\; r+1,\ldots,n$$

is sufficient. Unfortunately it has been observed that the resulting values of σ are very large and no successful algorithm of this type has been obtained. For more about how to choose the penalty parameter σ see Fletcher [1987] Chapter 12.

Algorithm 1.7.3 which has better convergence properties is now recommended. This differs from the formulation given in (5.3.25). An equivalent form to (1.7.12) is the following

$$\begin{aligned} & \underset{\boldsymbol{\delta}}{\text{minimize}} \quad \psi^{(k)}(\boldsymbol{\delta}) \\ & \textit{subject to}\;\; \mathbf{x}^{(k)} \;+\; \boldsymbol{\delta} \;\leq\; \mathbf{v} \\ & \qquad\qquad\quad \|\boldsymbol{\delta}\|_\infty \;\leq\; \rho^{(k)} \end{aligned} \qquad (5.3.26)$$

where

$$\psi^{(k)}(\boldsymbol{\delta}) \;=\; f^{(k)} \;+\; \nabla f^{(k)T}\boldsymbol{\delta} \;+\; \tfrac{1}{2}\,\boldsymbol{\delta}^T W^{(k)}\boldsymbol{\delta} \;+\; \sigma\{\textstyle\sum_{\substack{i,j=r+1\\ i\neq j}}^{n} |d_{ij}^{(k)} \;+\; \nabla\; d_{ij}^{(k)T}\boldsymbol{\delta}|\} \qquad (5.3.27)$$

giving a correction vector $\boldsymbol{\delta}^{(k)}$, so that $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\delta}^{(k)}$. Also the Lagrange multipliers associated with each of the modulus terms in (5.3.27) become the elements of the matrix $\Lambda^{(k+1)}$ for the next iteration. The subproblem (5.3.26) can be solved by methods similar to those used in QP. The two methods (5.3.22) and (5.3.26) are equivalent when $\mathbf{x}^{(k)}, \Lambda^{(k)}$ are sufficiently close to $\mathbf{x}^*, \Lambda^*$ and σ is large enough.

In Han's method [1977] it is necessary for $\nabla^2\mathcal{L}^{(k)} \geq 0$ to hold, which excludes the possibility of an unbounded solution to (5.3.22). However in (5.3.26) it is not necessary to force $\nabla^2\mathcal{L}^{(k)} \geq 0$ to hold, and the choice $\sigma = 2\ \max_i\ x_i^{(k)}$ can be used.

The terms $|d_{ij}(\mathbf{x})|$ in (5.3.25) are not smooth and can cause slow convergence in practice. The second order correction is included to alleviate these difficulties. Let $\boldsymbol{\delta}^{(k)}$ be the solution of (5.3.26), then the second order correction is obtained by repeating (5.3.26) with some modification to (5.3.26), giving the subproblem

$$\begin{aligned} &\underset{\boldsymbol{\delta}}{\text{minimize}} \quad \psi^{(k)}(\boldsymbol{\delta}^{(k)}) \\ &\textit{subject to}\ \mathbf{x}^{(k)} + \boldsymbol{\delta} \leq \mathbf{v} \\ &\qquad\qquad\qquad \|\boldsymbol{\delta}\|_\infty \leq \rho^{(k)} \end{aligned} \qquad (5.3.28)$$

where

$$\psi^{(k)}(\boldsymbol{\delta}^{(k)}) \;=\; f^{(k)} \;+\; \nabla f^{(k)T}\boldsymbol{\delta} \;+\; \tfrac{1}{2}\,\boldsymbol{\delta}^T W^{(k)}\boldsymbol{\delta} \;+\; \sigma\{\textstyle\sum_{\substack{i,j=r+1\\ i\neq j}}^{n} |d_{ij}^{(k)} \;+\; \nabla\; d_{ij}^{(k)T}\boldsymbol{\delta} \;+\; \gamma^{(k)}|\}$$

and

$$\gamma^{(k)} \;=\; \tfrac{1}{2}\;\boldsymbol{\delta}^{(k)T}\nabla^2\, d_{ij}^{(k)}\boldsymbol{\delta}^{(k)} \qquad (5.3.29)$$

and $\boldsymbol{\delta}^{(k)}$ calculated from (5.3.26). The solution to (5.3.28) is denoted by $\tilde{\boldsymbol{\delta}}^{(k)}$. The modified algorithm solves (5.3.26) as before to get $\boldsymbol{\delta}^{(k)}$ then calculates $\gamma^{(k)}$ using (5.3.29) then recalculates $\tilde{\boldsymbol{\delta}}^{(k)}$ using (5.3.28) and revised Lagrange multipliers $\tilde{\Lambda}^{(k+1)}$. Now $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \tilde{\boldsymbol{\delta}}^{(k)}$ and $\tilde{\Lambda}^{(k+1)}$ is used in place of $\Lambda^{(k+1)}$. Using the second order correction takes advantage of the readily available second derivative matrices $\nabla^2\, d_{ij} \quad i,j = r+1,\ldots,n$.

An important constraint has been neglected up till now, that is the variables $\mathbf{x} \in \Re^r$ must permit the matrix $\bar{A} + \textit{diag}\ \mathbf{x}$ to be factorized as in (1.5.4) with $D_1 > 0$. Therefore the restriction $D_1(\mathbf{x}) > 0$ on the feasible region of (5.3.10) is enforced. Also certain degenerate cases must be excluded. However if $\mathbf{x}^{(k)}$ is sufficiently close to $\mathbf{x}^*$ and r is identified correctly this restriction will usually be inactive at the solution. If $\mathbf{x}^{(k)}$ is remote from the solution then two constraints are introduced to avoid these disadvantages. Firstly the linearization of the constraint $d_{ii}(\mathbf{x}) \geq 0$

$$d_{ii}^{(k)} + \nabla d_{ii}^{(k)T} \boldsymbol{\delta} \geq 0. \quad i = r+1, \dots, n \tag{5.3.30}$$

are added to the subproblems (5.3.22),(5.3.26) or (5.3.28). Secondly the linearization of the constraint $D_1(\mathbf{x}) > 0$ about $\mathbf{x}^{(k)}$

$$d_{ss}^{(k)} + \nabla d_{ss}^{(k)T} \boldsymbol{\delta} > 0. \quad s = 1, \dots, r \tag{5.3.31}$$

However it is advisable not to allow $d_{ss}(\mathbf{x}^{(k)} + \boldsymbol{\delta})$ to become too close to zero, especially for small s which causes the factorization to fail ($D_1 \not> 0$). As a result the constraints

$$s\, d_{ss}^{(k)}/r + \nabla d_{ss}^{(k)T} \boldsymbol{\delta} \geq 0. \quad s = 1, \dots, r \tag{5.3.32}$$

are also included to the subproblems (5.3.22),(5.3.26) or (5.3.28).

Even with these extra conditions it might be difficult to find a partial factor for the matrix $\bar{A} + diag\, \mathbf{x}$ in the form (1.5.5) for some iterates $\mathbf{x}^{(k)}$. In this case smaller radius for the trust region is chosen with $\rho^{(k+1)} = \rho^{(k)}/4$, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ and $\Lambda^{(k+1)} = \Lambda^{(k)}$ are chosen for the next iteration.

Another restriction on the variables $\mathbf{x} \in \Re^r$ of (5.3.10) is that the bounds $x_i \leq v_i$, $i = r+1, \dots, n$ must remain inactive. This can be done by permuting the variables, although an acceptable permutation is not known in advance. Therefore the following procedure has been adopted. In the beginning of an iteration every variable is tested individually to reorganize the variables so that the active variables is first. As result of that the active bounds are those on variables $x_s^{(k)}$, $s = 1, \dots, p$ where p is number of active bounds. This permutation makes a complete change to the factorization (1.5.9) so that the matrix D_2 and the basis matrix Z are redefined. The Lagrange multipliers are reset to zero since they are not suitable to the redefined basis. Also the function $\phi(\mathbf{x})$ in (5.3.25) is redefined. The number of permutations made during the course of the algorithm must be finite, this is because the above procedures conflict with the global convergence strategy of reducing $\phi(\mathbf{x}^{(k)})$ monotonically if the number of permutations are not finite.

Another important consideration for the l_1 SQP method is how the integer r^* can be identified correctly. Since r^* is not known in advance it is necessary to estimate it by an integer denoted by $r^{(k)}$. Any change to $r^{(k)}$ causes a change to $\phi(\mathbf{x})$, and the number of variables in $\phi(\mathbf{x})$. It is important to consider the effect of making a fixed incorrect estimate r to r^*. If $r^{(k)} < r^*$ then the l_1 SQP method converges satisfactorily at a second order rate to a minimizer $\phi(\mathbf{x})$. Since r is too small this minimizer is not a solution to (5.3.10) because $d_{ij}(\mathbf{x}) \neq 0$ for some indices $i \neq j \quad i, j = r+1, \ldots, n$, and also because $\Lambda \geq 0$ does not usually hold. On the other hand if $r^{(k)} > r^*$ then the l_1 SQP algorithm converges to the minimizer of $\phi(\mathbf{x})$, which is the solution of (5.3.10) but the rate of convergence is very slow because the number of variables in $\phi(\mathbf{x})$ are increased. The slow rate of convergence indicates that the nonsmooth nature of the problem is not accounted for. The initial idea is to increase or decrease $r^{(k)}$ as the iteration proceeds, using the fact that $\Lambda^{(k)} \not\geq 0$ to increase $r^{(k)}$, and the existence of an active constraint for $s = r$ in (5.3.31) to decrease $r^{(k)}$. The above idea by Fletcher [1982] was not in fact investigated, which it may be necessary to do for large problems. However the more simple strategy described in Section 5.5 below proved to be very reliable and reasonably efficient, especially for $n \leq 20$.

Two examples for problem (5.3.1) are given which are similar to Examples 5.3.2 and 5.3.3.

Example 5.3.1

Consider problem (5.3.1) where

$$\bar{F} = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}.$$

The solution is $x^* = (3,\ 4/3,\ 3)$, no bounds are active i.e. $\boldsymbol{\pi}^* = 0$, and the set

$$K_{\Re} \cap K_{off}(\bar{F} + diag\,\mathbf{x}) = \{\bar{F} + diag\,\mathbf{x} | \begin{bmatrix} x_1 & 2 & 3 \\ 2 & x_2 & 2 \\ 3 & 2 & x_3 \end{bmatrix} \geq 0\} \qquad (5.3.33)$$

is illustrated in the neighbourhood of $\mathbf{x}^*$ in Figure 5.3.1

It can be observed that $K_{\Re} \cap K_{off}(\bar{F} + diag\,\mathbf{x})$ is convex but not a cone and is nonsmooth at $\mathbf{x}^*$. The rank of $F^* = \bar{F} + diag\,\mathbf{x}^*$ is $r = 1$, and its partial factors are

$$D = \begin{bmatrix} 3 & & \\ & 0 & \\ & & 0 \end{bmatrix} \qquad L = \begin{bmatrix} 1 & & \\ 2/3 & 1 & \\ 1 & 0 & 1 \end{bmatrix}$$

$$L^{-1} = V = \begin{bmatrix} 1 & \vdots & -2/3 & -1 \\ & \vdots & 1 & 0 \\ & \vdots & & 1 \end{bmatrix}.$$

thus

$$Z = \begin{bmatrix} -2/3 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The vector $\mathbf{b}^* = -2\mathbf{x}^* = (-6, -8/3, -6)$ satisfies (5.3.4a) and the corresponding $B^* \in \partial K_{\Re}$ is generated by the matrix

$$\Lambda^* = \begin{bmatrix} 8/3 & -8/9 \\ -8/9 & 6 \end{bmatrix}$$

($\Lambda^* > 0$ as required), and

$$B^* = -Z\Lambda^* Z^T = -\begin{bmatrix} 6 & -8/9 & -146/27 \\ -8/9 & 8/3 & -8/9 \\ -146/27 & -8/9 & 6 \end{bmatrix}.$$

Example 5.3.2

Another example for $n = 4$, let

Figure 5.3.1: The boundary of the restricted cone $(K_{\Re} \cap K_{off})(\bar{F} + \mathit{diag}\,\mathbf{x})$ in (5.3.33) (contours of x_2).

$$\bar{F} = \begin{bmatrix} 0 & 1 & 2 & -2 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ -2 & 2 & 1 & 0 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} 2 \\ 4 \\ 8 \\ 10 \end{bmatrix}.$$

The solution is $(2,\ 2.6505,\ 4.1209,\ 6.3537)^T$. The rank of $F^* = \bar{F} + \mathit{diag}\,\mathbf{x}^*$ is $r = 2$, and its partial factors are

$$D = \begin{bmatrix} 2 & & & \\ & 2 & & \\ & & 0 & \\ & & & 0 \end{bmatrix} \qquad L = \begin{bmatrix} 1 & & & \\ 0.5 & 1 & & \\ 1 & 1 & 1 & \\ -1 & 1.5 & 0 & 1 \end{bmatrix}$$

$$L^{-1} = V = \begin{bmatrix} 1 & -0.5 & \vdots & -2/3 & -1 \\ & 1 & \vdots & -1 & -1.5 \\ & & \vdots & 1 & 0 \\ & & \vdots & & 1 \end{bmatrix}.$$

thus

$$Z = \begin{bmatrix} -2/3 & -1 \\ -1 & -1.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The bound $x_1 \leq v_1$ is active and has a Lagrange multiplier $\pi_1^* = 55.37079$. The vector $\mathbf{b}^* + \boldsymbol{\pi}^* = -2\mathbf{x}^*$ satisfies (5.3.4a) and the corresponding $B^* \in \partial K_{\Re}$ is generated by the matrix

$$\Lambda^* = \begin{bmatrix} 8.2418 & -10.5108 \\ -10.5108 & 12.7074 \end{bmatrix}$$

If the bound v_1 is increased to $v_1 = 4$ for example, then the bound $x_1 \leq v_1$ becomes inactive and the vector $(2,\ 2.6505,\ 4.1209,\ 6.3537)^T$ is feasible but not optimal with $\sqrt{\mathbf{x}^T\mathbf{x}} = 8.269$. This time the conditions (5.3.4a) and (5.3.4b) do not hold. The optimal solution to this modified problem is

$$\mathbf{x}^* = (3.4555,\ 3.1833,\ 3.1833,\ 3.4555)^T$$

with $\sqrt{\mathbf{x}^T\mathbf{x}} = 6.644$ and $r = 2$. Second order conditions are used in this modified problem.

5.4 A hybrid method

In this section a new method for solving problem (5.1.2) is considered. The method described here depends upon both projection and l_1 SQP methods using a hybrid method. The projection method which converges globally but often converges at very slow order. Meanwhile in the l_1 SQP method which converges at second order if the correct rank r^* is given. The main disadvantage of the l_1 SQP method are that they require the correct r^*. The projection–l_1 SQP method starts with the projection method to determine the rank $r^{(k)}$ and continues with the l_1 SQP method.

The method in this section follows a similar strategy to that in Section 4.3. Since r^* 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. This is because the projection method is a globally convergent method.

Consider Λ_r from (5.2.4) then at the solution the number of eigenvalues in Λ_r is equal to the rank of A^*. Thus

$$No.\ \Lambda_r^* \ = \ rank(A^*) \ = \ r^* \tag{5.4.1}$$

where $No.\ \Lambda$ is the number of positive eigenvalues in Λ. A similar equation to (5.4.1) is used to calculate an estimated rank $r^{(k)}$ and is given by

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

where Λ_r is given by (5.2.4). The range of error is relatively small. The l_1 SQP method will be applied to solve the problem as described in Section 5.3.

The projection–l_1 SQP algorithm can be described as follows.

Algorithm 5.4.1

Given any matrix $F \ = \ F^T \ \in \ \Re^{n\times n}$, let s be a positive integer. Then the following algorithm solves problem (5.1.2)

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

ii. Apply the projection method until

$$No.\ \Lambda_r^{(k)} \ = \ No.\ \Lambda_r^{(k+j)} \qquad j \ = \ 1, 2, \ \ldots, s \tag{5.4.2}$$

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

iv. Use the result vector $\mathbf{x}$ from projection method as an initial vector for the l_1 SQP method

v. Apply the l_1 SQP method for solving problem (5.1.2).

The integer s in Algorithm 5.4.1 can be any positive number. If it is small then the rank $r^{(k)}$ may not be accurately estimated, however the number of iterations taken by the projection method is small. In 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_1 SQP method is that if $F^{(0)}$ is positive semi–definite (singular) then the projection method terminates at the first iteration. Moreover it gives the best estimate to $r^{(k)}$.

It has been found difficult to produce an algorithm starting with l_1SQP method and then using the projection method to update the rank, in contrast to the method in Section 4.4.

A way of finding a lower bound on the rank $r^{(k)}$ is suggested by Fletcher [1985]. The number of free variables in problem (5.1.2) are at most n, and this can be reduced to $n - p$ if there are p active bounds at the solution. Since $D_2 \in \Re^{(n-r)\times(n-r)}$ and symmetric then the equation $D_2 = \mathbf{0}$ introduces $1/2\,(n - r + 1)(n - r)$ conditions, so except in degenerate cases it follows that

$$n - p \geq 1/2\,(n - r + 1)(n - r) \tag{5.4.3}$$

which imposes a significant restriction on the dimensions of D_2. For example if $n - p = 20$ and $n = 21$ then r can be no smaller than 14.

5.5 Numerical results and comparisons

In this section numerical examples are given for the projection algorithm l_1 SQP algorithm and Algorithm 5.4.1. First numerical examples for Algorithm 5.2.2 are given in some detail in Table 5.5.1 then the same numerical examples for l_1 SQP algorithm and Algorithm 5.4.1 are given in Table 5.5.2.

The numerical test problems are obtained from the data given in Table 6.2.1, by Woodhouse [1976].

The projection Algorithm 5.2.2, l_1 SQP algorithm and Algorithm 5.4.1 are applied to solve problem (5.1.2). The Woodhouse data set is a 64×20 data which corresponds to 64 students and 20 subtests. 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 5.5.1 and 5.5.2, the value of n is the number of elements in each subset.

The results obtained by the Algorithm 5.2.2 are tabulated in Table 5.5.1. 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 5.5.1 the column headed by NI gives the number of iterations used by the projection method. It is clear from Table 5.5.1 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 5.5.1 gives the correct rank r^* for each particular problem. The order of convergence is very slow as can be seen from Table 5.5.1. Also in Table 5.5.1 the optimal x_i^* for $i = 1, 2, ..., n$ and $\|\mathbf{x}^*\|_2$ are given. Finally, the eigenvalues for the projection method are solved using the NAG library.

At the end of Section 5.3 a difficult strategy had been described for applying the l_1 SQP method. A more simple strategy has been adopted. Initially choosing $r^{(k)}$ as the smallest integer compatible with (5.4.3). Starting from $\mathbf{x}^{(0)} = \mathbf{v}$, $\Lambda^{(0)} = 0$ and $\rho^{(0)}$ supplied by the user then $\phi(\mathbf{x})$ is minimized by the iteration based on (5.3.26) as described in Section 5.3. Thus if $\|D_2(\mathbf{x})\| \leq \epsilon$ for some small ϵ, at the solution then the algorithm terminates. If not then $r^{(k)}$ is increased by one. Then a new variable x_{r+1} is adding to problem (5.3.10). This variable is estimated by adding the value of the l_1 norm of the first column of D_2 to the current value of x_{r+1} as given by (5.3.9). Then the partial factors of the new matrix are well–determined. Also increases in $r^{(k)}$ reduce the dimension of D_2. The Lagrange multiplier matrix is changed by deleting all the elements in the first row and column. The radius $\rho^{(k)}$ is reinitialized and finally the iteration based on (5.3.26) is used to solve this problem. After a few repetitions r^* will be identified.

In Table 5.5.2 three methods are compared: projection method (PM), l_1SQP algorithm and projection–l_1SQP algorithm (Pl_1SQP). The stopping criterion is $\|\mathbf{x}^{(k+1)}\| - \|\mathbf{x}^{(k)}\| < 10^{-8} = \epsilon$. It is estimated that x_i are accurate to $4 - 5$ decimal places and $\|\mathbf{x}\|$ is accurate to $6 - 7$ decimal places. In Table 5.5.2 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 problem (5.3.26) is solved. $r^{(0)}$ in the column headed by l_1SQP gives the initial rank for F using equation (5.4.3) and $r^{(0)}$ in the column headed by Pl_1SQP gives the initial rank for F using Algorithm 5.4.1. The three

Columns which determine F	r^*	NI	$x_i^* \quad i = 1, 2, .., n$				$\sqrt{\sum (x_i^{2*})}$
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 5.5.1: Results for problem (5.1.2) from projection Algorithm 5.2.2.

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 $\bar{A} + \mathit{diag}\,\mathbf{x}^*$ is exactly singular and positive semi–definite for all methods. The initial value of $\rho^{(0)}$ is 20.0. In l_1SQP most cases require a few iterations for solving (5.3.10) as r increases. For each value of r second order convergence of the iteration based on (5.3.26) and (5.3.28) is obtained.

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. In the Pl_1SQP algorithm the initial value of $\rho^{(0)}$ is 5.0.

Finally the projection method is not very successful in estimating the rank r^* especially when $n \geq 12$ and a more effective method is required to give a better estimate for r^* similar to those methods in Chapter 4.

Columns which		PM	l_1SQP		Pl_1SQP		
determine A	r^*	NI	$r^{(0)}$	NQP	NI	$r^{(0)}$	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 5.5.2: Numerical comparisons of methods of this chapter.