Computing the Nearest Correlation Matrix using Difference Map Algorithm

by

Xuezhi Cui

A research paper
presented to the University of Waterloo
in partial fulfillment of the requirements for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Prof. Henry Wolkowicz

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The *difference map algorithm* (DMA) is originally designed to find the global optimal solution to nonconvex problems. The main feature of DMA is that it can avoid the *stagnation*, (which always occurs when applying the *alternating projection method* (APM) on nonconvex problems so that APM is trapped at local minimized point and fail to find the global optimal solution.) and converge to the global solution directly. We show that DMA is not only a better algorithm to solve nonconvex problems but also is better to solve convex problems. We using the *nearest correlation matrix* to show this statement. The *nearest correlation matrix* (NCM) problem is to find the nearest point to a given symmetric matrix A in the intersection of positive semi-definite cone and the unit diagonal space. This is a very typical example of convex problems. Currently, three popular ways to solve this problem are: the *alternating projection method*; *(quadratic) Newton's method*; and the *primal-dual interior-exterior-point approach.* The least expensive of these methods is the alternating projection method. However, this method can only achieve a linear rate of convergence and may cause a huge computational cost when we are looking for a highly accurate solution. Because of the similarity of DMA and APM, we are going to compare the best results obtained by both algorithms and the computational cost with different desired accuracies of the final solution. Numerical experiments show that DMA with β value equals to 1 can obtain the best solution and it is identical to the one obtained by APM. However, when A is large sized and ill-conditioned, or we are seeking a highly accurate solution, DMA can find the optimal solution with less computational cost.

Keywords: correlation matrix; alternating projection method; weighted Frobenius norm; difference map method; phase retrieval problem, nonconvex problems.

Acknowledgements

I would like to thank Professor Henry Wolkowicz for all of his guidance and help that make this project possible.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

A *correlation matrix of order n*, $X \in \mathcal{C} \subset \mathcal{S}^n_+$, is a symmetric $n \times n$ positive semi-definite matrix such that all elements on the diagonal are 1. Correlation matrices have wide applications in the financial industry. For example, according to one of the most famous theories of investment, Markowitz's modern portfolio theory (MPT), to find the so-called efficient frontier we first need to build up the correlation matrix for the assets in the portfolio, then we use the corresponding quadratic function with Hessian X to balance off maximizing the expected return while minimizing the corresponding risk level, e.g., [16, 17, 18]. Also in the process of finding the so-called *Sharpe ratio*, the correlation matrix plays a key role [23].

However, in reality we might not have a complete collection of the data we need. For example, among a stock's last 10 years daily returns, there is a large possibility that we might not know many data elements in the matrix and/or some of the data elements are noisy (are in error). Thus, to build up the correlation matrix, we have to approximate the missing correlations and/or find an approximate correlation matrix. Moreover, one missing data element does not only affect the corresponding entry in the correlation matrix, but also the whole corresponding row and column in that matrix [15]. Since some of the entries in the correlation matrix are specified and others are not, this kind of matrix is called a *partial symmetric matrix* . To apply the MPT or calculate the Sharpe ratio, we have to solve the corresponding (approximate) matrix completion problem [10]. Moreover, to minimize the error in future calculations, we need to complete the correlation matrix such that the distance between the original partial matrix and the completed matrix is a minimum. This completed matrix is called the *Nearest Correlation Matrix (NCM).*

Currently, there are three methods used to solve the NCM problem. *alternating projection method*(APM), *quadratically convergent Newton method* and *primal-dual interior-exterior-point*(p-d i-e-p). p-d i-e-p can find the nearest correlation matrix with the highest Either of them has both advantages and drawbacks. we compare them in Section 1.2

In this thesis, we introduce a new algorithm to solve NCM problem. It is called the *difference map algorithm*(DMA), which is popularized by Veit Elser. Similar to APM, DMA involves only the projection operations plus a simple real parameter β. DMA is designed initially to find the global optimal solution to the nonconvex problems. General results show that DMA is a very useful algorithm to solve nonconvex problems [22]. It can easily avoid the local traps and directly converge to the global optimal points with a relatively small number of iterations. (We use the example in [22] to illustrate that statement in Chapter 4.) However, DMA with different β values may lead us to different results for the same starting point. In Section 3.1, we run several test to determine the β value that gives us the best solution, and the test results show that the best solution obtained by DMA when β is 1, and it is actually the same point obtained by APM. Because of the similarity of DMA and APM, we mainly compare how efficiently that DMA can find the optimal solution comparing to APM. Numerical experiments in Section 3.2 show that even though the best results obtained by DMA has the same distance between the solution and the original matrix as APM, but DMA can achieve the higher accuracy (i.e higher *tolerance*) solution than APM more quickly, and has higher performance When the original matrix is large and ill-conditioned.

1.1 Nearest Correlation Matrix

The NCM problem with the Frobenius norm is: given an arbitrary $n \times n$ symmetric matrix $A \in \mathcal{S}^n$, find a nearest correlation matrix, i.e, solve the program

$$\begin{array}{cc} \min & \frac{1}{2}\|A - X\|_F^2 \\ \text{s.t.} & \operatorname{diag} X = e \\ & X \succeq 0, X \in \mathcal{S}_+^n \,. \end{array} \tag{1.1}$$

Here e is a vector of all ones; $\mathcal{S}_+^n$ is the convex cone of $n \times n$ positive semi-definite symmetric matrices in the space of symmetric matrices, $\mathcal{S}^n$, equipped with the trace inner-product $\langle A, B\rangle = \text{trace } AB$; and, we have the corresponding Frobenius norm is $\|A\|_F^2 = \sqrt{\langle A, A\rangle}$. We let $\operatorname{diag} X$ denote the vector formed with the elements from the diagonal of X. Choosing the Frobenius norm in (1.1) because that it is the easiest norm to work [12]. Often in these nearest approximation problems, we use a weighted Frobenius norm for scaling purpose. The two most common norms are W-norm and H-norm. The NCM problem 1.1 in W-norm is defined as

$$\begin{array}{cc} \min & \frac{1}{2}\|W^{1/2}(A - X)W^{1/2}\|_F^2 \\ \text{s.t.} & \operatorname{diag} X = e \\ & X \succeq 0, X \in \mathcal{S}_+^n \,. \end{array} \tag{1.2}$$

where $W \in \mathcal{S}^n$ is a positive definite matrix, and the NCM problem 1.1 in H-norm

$$\begin{array}{cc} \min & \frac{1}{2}\|H \circ (A - X)\|_F^2 \\ \text{s.t.} & \operatorname{diag} X = e \\ & X \succeq 0, X \in \mathcal{S}_+^n \,. \end{array} \tag{1.3}$$

Here, $\circ$ denotes the Hadamard product, i.e, $A \circ B = (a_{ij}b_{ij})$, and all elements in $H \in \mathcal{S}^n$ are positive. The motivation for using weighted norms during the computation is to reduce the error coming from the estimation of the missing entries and make the prediction more precise. Note that both W and H are not any random matrix. The choice of the weights depends on the noise model we used for solving the NCM problems. The two most popular models are multiplicative or additive noise model. The details about how the noise model affects the choice of the weight are in e.g., [13]. Briefly, we can assign a weight to a single entry in A when using (1.3), and the magnitude that we assign usually depends on how accurately we know the corresponding entry a_{ij} in A. The expression inside the norm in (1.2) is a congruence and preserves inertia, see e.g. [12]. However, this means that we cannot directly (simply) put specific weights on individual elements in A. Generally, (1.2) is easier to work with during the computation of projections in the algorithms, because we can apply spectral decompositions (See Section 2.2.1). In this thesis we mainly discuss the NCM problem in form (1.2).

1.2 Three Current Algorithms for solving NCM problems

Currently, there are three main algorithms to find the NCM. The first one is using the *Alternating Projection Method (APM)*, which is described by Higham [12]. Based on Dykstra's work, Higham applied APM to find the solution to the NCM problem 1.1 in 2001. His algorithm guarantees the convergence to the optimal solution and the result obtained by him makes it possible to deal with NCM problems in practice. However, the maximum convergence rate we can get from APM is linear [12], and the speed of convergence depends on the angle between the two constraint sets [7]. To reach a high accuracy of the solution, APM may take a long time and have a huge computational cost if the angle between the two sets is very small.

The second approach uses the *quadratically convergent Newton's method*, which is developed by Qi and Sun in 2005 [20]. Since the dual of the object function involves the projection onto the semidefinite cone, it is not twice continuously differentiable. They use the strong semi-smoothness of the metric projector in order to apply Newton's method [20]. This method increases the speed of convergence up to the quadratical rate, which is much higher than the rate for APM. On the other hand, if we require the accuracy of solution to be high (e.g., $error \approx 1 \times 10^{-8}$), this method may fail to converge to the optimal points [20]. Moreover, applying the Newton's algorithm is relatively more complicated. It may be easier to use APM to solve NCM problem in practise.

Anjos et al, [1] treat the NCM problem as a semi-definite programming problem and use a primal-dual interior-exterior-point (p-d i-e-p) approach to solve the problem. This approach can always converge to a optimal point in the intersection of contraints [1, Thm 3.2], and the accuracy of the solution that p-d i-e-p can achieve is the highest among all three methods, even when the original matrix is highly degenerate[1]. However, although this algorithm can work efficiently when A has a large dimension and sparse, the computational cost is still very expensive.

Also, quadratically convergent Newton's method and p-d i-e-p are specially designed to solve NCM problems. APM can be applied to other problems, but can only achieve local convergence if the problem is nonconvex. Comparing to those three methods, DMA can be applied to solve much more problems. DMA is originally developed to find the global optimal solution to nonconvex problems. However, we show that DMA can also be applied to solve convex problems and its performance is even better.

Chapter 2

Theory

DMA is a Metaheuristic algorithm, which has two basic meanings. The first one is that it is a computational iterative algorithm that searching for the optimal solution by performing the projections onto both constraints. The second one is that it requires few or no assumptions when we apply it to solve problems. Thus, there is some similarities between DMA and APM. (Both involve the projections onto the constraint sets, both requires few assumptions.) To understand how DMA works, we first start by considering APM.

2.1 Alternating Projection, APM, Type Methods

The idea of using alternating projection method to find a point in the intersection of closed subspaces was introduced by von Neumann in 1933, see e.g., the survey [3, Thm. 13.7] and the paper [24]. In this section we consider APM with the corrction by Dijkstra [8] and Higham [12].

For our NCM problem, we define APM as:

$$X_k \leftarrow P_U(P_S(X_{k-1})), \tag{2.1}$$

where S and U are the two closed convex constraint sets. S is the positive semi-definite cone which is defined as:

$$S = \{X \in \mathcal{S}^n : X \succeq 0\}. \tag{2.2}$$

Here X is a symmetric matrix. The other constraint set U is the set of matrices with diagonals all ones:

$$U = \{X \in \mathcal{S}^n : \operatorname{diag}(X) = e\}. \tag{2.3}$$

where $e \in \mathbb{R}^n$ is a vector of all ones.

It is shown that if the two sets are closed subspaces of a Hilbert space, then APM is convergent to the nearest point in the intersection of these two constraint sets [5]. However, our two sets are closed convex sets instead of closed subspaces. Although it is known that APM in the form of formula (2.1) can definitely reach one point in the intersection if the constraints are closed convex, the algorithm may not converge to the optimal point X, which is the nearest point in the intersection space to the original matrix A, see e.g., [11]. To ensure that the APM converges to the optimal point, i.e, the nearest correlation matrix. Dykstra adds a small correction on formula (2.1):

$$X_k \leftarrow P_U(P_S(X_{k-1} - I_{mod_{k-1}})), \tag{2.4}$$

where $I_{mod_{k-1}}$ is the correction to the current projection in the $k-1$ iteration . It is calculated as a normal vector to the corresponding convex set [12]. APM defined in formula (2.4) is guaranteed to converge to the optimal solution.

Theorem 2.1.1. *[8, Thm 3.1] Suppose U and S are closed convex sets, then the matrix X_k defined in* (2.4) *converges to the true solution X^* of problem 1.1 as $k \to \infty$.*

The detailed proof of this theorem is listed in the appendix in [8], also it is mentioned in [6] and [4, Thm. 2]. APM algorithm based on Definition 2.4 is described in Algorithm 2.1.1 The stopping condition is either that the iteration step k has reached the upper limit (denoted as *maxit*), or the error $\Delta \leq$ tolerance(denoted as *tol*). The error Δ is defined as [12]:

$$\Delta = \max\{\tfrac{\|X_k - X_{k-1}\|_\infty}{\|X_k\|_\infty}, \tfrac{\|X_k^* - X_{k-1}^*\|_\infty}{\|X_k^*\|_\infty}, \tfrac{\|X_k - X_k^*\|_\infty}{\|X_k\|_\infty}\}. \tag{2.5}$$

Note that tolerance determines the accuracy of final solution.

```
Algorithm 2.1.1: APM algorithm [12]
input  : Symmetric $A$ in $\mathcal{S}^n$
output: The nearest correlation matrix, $X$
begin
    Initialization: $X_0 = A, k = 1, I_{mod_0} = 0, \Delta = 1$;
    while $\Delta \geq tol$, or $k \leq maxit$ do
        $R_k = X_{k-1} - I_{mod_{k-1}}$, % $I_{mod_{k-1}}$ is Dykstra's correction;
        $X_k^* = P_S(R_k)$;
        $I_{mod_k} = X_k^* - R_k$;
        $X_k = P_U(X_k^*)$;
    end
end
```

From the definition 2.4, we see that in every iteration of APM, the output is a projection onto the constraint U. Thus the final optimal solution obtained from APM is also in the subspace U. APM is a simple iterative algorithm (i.e, no assumptions, only few projection operations involved), so it is easy to apply in practice. However, the drawback of APM is also obvious: that is, the rate of convergence. Deutsch and Hundal [7] state that the convergent speed of APM is related to the angle between the closed subspaces. (Let's define this angle as α.) Since there are only two closed convex constraint sets, to see how the rate of convergence is related to α, we first define the upper bound of the global error of APM as:

$$E = \|(P_U P_S)^k(A) - P_{U\cap S}(A)\|, \tag{2.6}$$

where n is the numbers of iteration. Then according to Aronszajn's result[2, Corollary 2.9], the upper bound E in (2.6) is less than or equal to the cosine of the angle between the two constraints U and S

Corollary 2.1.1. *[2, Corollary 2.9], For the pair of closed convex sets U and S, the upper bound of the global error of APM is $\|(P_U P_S)^k(A) - P_{U\cap S}(A)\| \leq c(U,S)^{2k-1}$, for all $k \geq 1$, where $c(U,S)$ is the cosine of the angle α between the two constraint sets U and S.*

The detailed proof is in [2]. From the Corollary 2.1.1, we can see that if α is small, then cosine of α is close to 1. There is a possibility that APM takes sufficiently large number of iterations to converge to the optimal point, if we require the solution to be highly accurate.

2.2 Difference Map Algorithm, DMA

In this section, we show how DMA is defined and its convergence. Also, to build up DMA algorithm that we are using in numerical experiments, we first develop the algorithms used to find the projections onto both constraints.

For any $k \geq 1$, one iteration of DMA can be seen as [9, Equation 16]:

$$X_k = X_{k-1} + \overrightarrow{\mathbf{d}}, \tag{2.7}$$

where the direction $\overrightarrow{\mathbf{d}}$ is defined as:

$$\overrightarrow{\mathbf{d}} = \beta\left\{P_S\left[(1+\beta^{-1})P_U(X_{k-1}) - \beta^{-1}X_{k-1}\right] - P_U\left[(1-\beta^{-1})P_S(X_{k-1}) + \beta^{-1}X_{k-1}\right]\right\}. \tag{2.8}$$

Here β can be any real number excluding 0, (Studies show that DMA with the β value may still converge to a fixed point but with a huge number of iterations and the solution is not optimal [9].) X_k is the current point at the beginning of the iteration, P_S and P_U are two projections of X_k onto the constraints S and U, respectively. These two projections are the same as the projections we used in APM. (details will be discussed later). The solution X is obtained when $\overrightarrow{\mathbf{d}} = 0$. The algorithm for one iteration of DMA is stated in Algorithm 2.2.3.

Theorem 2.2.1. *[9, Section 4], suppose U and S are closed convex sets, and there exists a minimum local separation δ_{min} of constraints U and S, then the matrix X_k which is defined in (2.7) is convergent to a point in the intersection of U and S, say $X_{U\cap S}$ as $k \to \infty$.*

The details of proof are in [9]. Since there is barely no assumption for DMA to achieve convergence, and it iterations only involves projections and one real parameter, DMA is a very simple method to apply in practice, compare to the Newton's method and p-d i-e-p method described in Section 1.2. Based on Theorem 2.2.1, we can only conclude that DMA is able to converge to the point in the intersection of both constraints. This point may or may not be the optimal one, i.e, the nearest correlation matrix to A. In fact, with different β values, DMA may converge to different points in the intersection. Therefore, we need to find which β value gives us the point that is nearest to the original matrix A before we compare the efficiency and accuracy of DMA and APM. We use numerical experiments to determine the best β values, and the details and results are in Section 3.1. To make the numerical experiments run, we first describe the algorithms used to find projections onto both constraints and the DMA algorithm.

2.2.1 Projections

Now let us talk about the algorithms using to find the two projections onto U and S. Remember that we work with the NCM problem in the weighted norm, which is defined in (1.2). The following results are based on Higham's work [12], since the set of problems and constraints are the same.

First of all, we need to define several notations. For any two matrices A and B, their weighted inner product is defined as

$$\langle A, B\rangle_W = \operatorname{trace}(A^T WBW). \tag{2.9}$$

For any convex set $C \subset \mathcal{S}^n$, the normal cone at $X^* \in C$ is

$$N_C(X^*) = \{V \in \mathcal{S}^n : \langle V, X - X^*\rangle \leq 0, \text{ for all } X \in C\}. \tag{2.10}$$

From the Definition 2.9 and 2.10, we have the following two results:

Lemma 2.2.1. *[12, lemma2.1], for $A \in U$,*

$$N_U(A) = \{W^{-1} \operatorname{diag}(\theta_i) W^{-1} : \theta_i \textit{ arbitrary}\} \tag{2.11}$$

Lemma 2.2.2. *[12, lemma2.2], for $A \in S$,*

$$N_S(A) = \{V \in \mathcal{S}^n :< V, A >= 0, V \preceq 0\} \tag{2.12}$$

The proofs are in [12].

We can use the normal cone to find the projections onto the set U and S. Let us denote the projection onto U as P_U.

Theorem 2.2.2. *[12, Thm 3.1]. For the W-norm, the projection of A onto set U is*

$$N_U(A) = A - W^{-1} \operatorname{diag}(\theta_i) W^{-1}. \tag{2.13}$$

and the $\theta = [\theta_1, ..., \theta_n]^T$ is the solution to the linear system

$$W^{-1}\theta W^{-1} = \operatorname{diag}(A - I). \tag{2.14}$$

Algorithm 2.2.1: `Projection onto Constraint` U `in MATLAB`

```
input  : Symmetric $A$ in $\mathcal{S}^n$
output: The projection of $A$ onto the constraint set $U$, $P_U$
begin
    $P_U = A$;
    if $i = j$ then
        $P_{U_{ij}} = 1$;
    else
        $P_{U_{ij}} = P_{U_{ij}}$;
    end
end
```

In MATLAB, we can follow the algorithm 2.2.1 to find the projection onto U.

It is more difficult to find the projection onto constraint S. Higham developed the closed formula only for W-norm [12]. There is no such one for H-norm, because we can not apply the spectral decomposition on H-norm. In the later research, Qi and Sun introduced an augmented Lagrangian dual approach to solve the H-norm NCM problem [21]. To show the formula, we need to denote several notations first.

Let $D = \mathrm{diag}(\lambda_i)$. Then the spectral decomposition of $A \in \mathcal{S}^n$ is

$$A = QDQ^T \tag{2.15}$$

where Q is orthogonal. Then we denote that

$$A_+ = Q\,\mathrm{diag}(max(\lambda_i, 0))Q^T \tag{2.16}$$

Then, we have the projection onto constraint S in the W-norm.

Theorem 2.2.3. *[12, Thm 3.2], for NCM problem in the weighted norm defined in 1.2,*

$$P_S(A) = W^{-1/2}((W^{1/2}AW^{1/2})_+)W^{-1/2} \tag{2.17}$$

Moreover,

$$\mathrm{diag}(P_S(A)) \succeq \mathrm{diag}(A). \tag{2.18}$$

The detail of proof is in [12]. Based on the equation (2.17), we can derive the computational algorithm for the projection onto S.

2.2.2 The DMA Pseudocode

Now we have found all the elements which we need to solve the NCM problem (1.1) by DMA. We list the steps in algorithm (2.2.3) that we code in MATLAB. Assume that the β value is

Algorithm 2.2.2: `Projection onto Constraint` S `in MATLAB`

```
input  : Symmetric $A$ in $\mathcal{S}^n$
output: The projection of $A$ onto the constraint set $S$, $P_S$
begin
    $[V, \lambda] = eig(A)$;
    if $\lambda_i \geq 0$ then
        $\lambda_i = \lambda_i$;
    else
        $\lambda_i = 0$;
    end
    $P_S = V\lambda V^T$;
    $P_S \leftarrow \frac{1}{2}(P_S^T + P_S)$ (to ensure symmetry)
end
```

determined already.

Algorithm 2.2.3: `the Difference Map Algorithm`

```
input  : Symmetric $A$ in $\mathcal{S}^n$, $\beta \in \mathbb{R}$ ;
output: a estimated nearest correlation matrix $X$ ;
begin
    while $\Delta \geq tol$ and $k \leq maxit$ do
        $f_S(X_k) = P_S(X_k) - (P_S(X_k) - X_k)/\beta$;
        $f_U(X_k) = P_U(X_k) + (P_U(X_k) - X_k)/\beta$;
        $X_{k+1} = X_k + \beta[P_U(f_S(X_k)) - P_S(f_U(X_k))]$;
        $k = k + 1$;
    end
end
```

Here the projections P_U and P_S can be found from Algorithms (2.2.1) and (2.2.2) and the stopping condition is either that the iteration step k has reached the upper limit (denoted as $maxit$), or the distance between these two projections $\Delta \leq$ *tolerance* (denoted as *tol*), where

$$\Delta = \|P_U(f_S(X_k)) - P_S(f_U(X_k))\|_F \leq tol. \tag{2.19}$$

Note that tolerance determines the accuracy of final solution.

Chapter 3

Computations

3.1 The best β value

Numerical results show that DMA with different β values may converge to different points in the intersection. (Figures of results are in Chapter 6.) It means that DMA can always converge to a correlation matrix of A but that matrix may not be the nearest correlation matrix. Therefore, to solve NCM problem 1.1, we need to find out the β value with which DMA can give us the nearest correlation matrix. Solving this problem contains two steps, we first find the minimized distance among all distances obtained by DMA with different β values, and its corresponding β value. Then we compare this minimized distance given by DMA with the solutin obtained by APM. According to Theorem 2.1.1, We recall that the APM Algorithm 2.1.1 is always convergent to the nearest correlation matrix to A. Thus, if the minimized distance obtained by DMA is equal to that obtained by APM, we can conclude that DMA is able to find the nearest correlation matrix of A as well.

We run the numerical experiments using MATLAB $R2009a$. The unit roundoff is approximately $u \approx 1.1 * 10^{-16}$. The computer we use to run the program is equipped with a Intel I5 2.8GHZ CPU and 4GB DDR3 RAM. For those numerical experiments in this section, we set the tolerance (tol) to be 1×10^{-8} and the maximum numbers of iteration (maxit) to be 100. For the simplicity, we set the weighted matrix W to be the unit diagonal matrix.

In numerical experiments, we first need to do is to create a random symmetric matrix A. This matrix does not need to be positive semi-definite. We use the command *sprandsym(n,density,rc)*, which generates a $n \times n$ symmetric random sparse matrix with elements between -1 and 1. We choose the dimension n of A from 10 to 100 to cover most of the cases in reality. The value of variable*density* is between 0 and 1. It determines the number of nonzero entries in the sparse matrix by the formula [19]

$$\#\text{ of nonzero elements in } A = density \times n \times n. \tag{3.1}$$

The closer to 0 the density is, the more zero elements the matrix has, and conversely, there is more non-zero elements generated in A as density is closer to 1. The variable rc is short for reciprocal condition number of the matrix, or 1/(condition number). Therefore the range of variable $rc \in (0, 1]$. However, only the unit diagonal matrix (i.e, identity matrix) can have the condition number equal to 1. Having a correlation matrix which is a identity matrix means that all stocks in the portfolio are uncorrelated. That is impossible to happen in practice. Moreover, we should expect the condition number of the correlation matrix in practice to be large, i.e, rc is small(e.g., $rc \leq 0.5$). Based on the help documents in MATLAB, using of rc can generate a random symmetric matrix A with elements between -1 and 1, and the condition number of A is equal to $1/rc$ [19].

Next, We need to find out which β value can let DMA converge to the nearest point in the intersection to A. The testing range of β values is set to be between -2 and 2. We run the test for few cases with different input variables n, density and rc. For each case, we run the numerical experiment 100 times based on Algorithm 2.2.3 with every β value in the range, and record only the one that gives us the minimized distance. In case that more than one β values make DMA converge to the nearest point, we take the largest β value to record. Therefore, there are total 100 results in each case and we show all the results in histograms in Figures from 3.1 to 3.4.

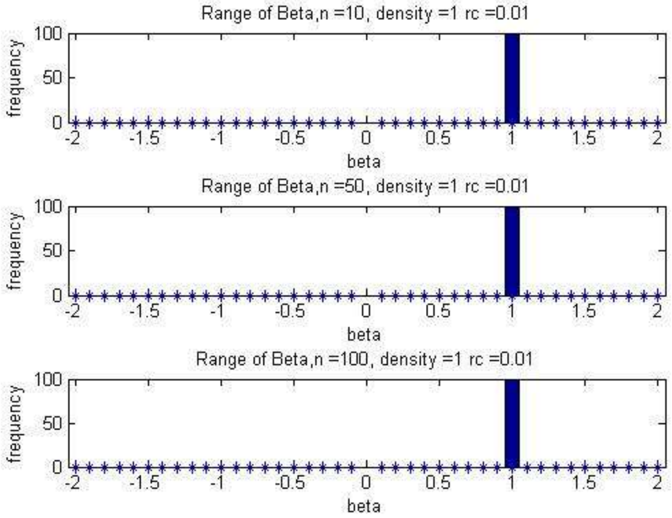

Figure 3.1: β values which give us the minimized distance, $density = 1$, $rc = .01$

Clearly, the nearest point is obtained when β is equal to 1 for all cases. Therefore, we would like to set β value to be 1 to run later numerical experiments. Also note that when β is 1, we can reduce the equation (2.8) to a simpler form which has less projection operations involved in each iteration:

$$\overrightarrow{\mathbf{d}} = P_S(2P_U(X_{k-1}) - X_{k-1}) - P_U(X_{k-1}). \tag{3.2}$$

Because of the reduction of projection operations involved, the time that one iteration of DMA algorithm takes in the form of Equations (3.2) is expected to be less when solving the

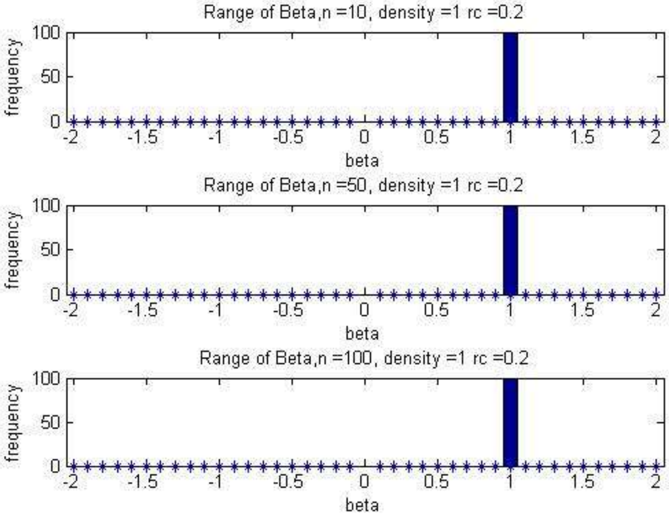

Figure 3.2: β values which give us the minimized distance, $density = 1$, $rc = .2$

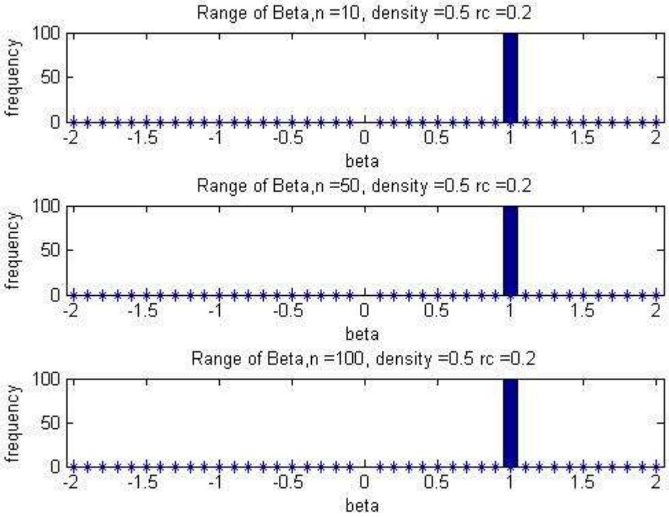

Figure 3.3: β values which give us the minimized distance, $density = .5$, $rc = .2$

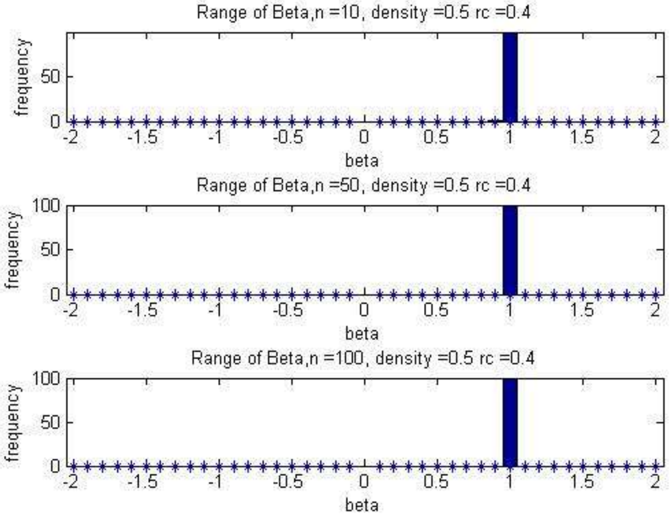

Figure 3.4: β values which give us the minimized distance, $density = .5$, $rc = .4$

NCM problem (1.1). The rounding errors which occur during the searching process is also deducted.

The next test we need to do is to determine if the optimal point obtained by DMA with $\beta = 1$ is the nearest correlation matrix. Same as precious test, we test for several cases. For each cases, we run numerical experiments 100 times and record the average of the minimized distances obtained by both algorithms. The results are shown in Tables from 3.1 to 3.4.

n	DMA, $\beta = 1$	APM
10	3.0990	3.0990
50	7.0903	7.0903
100	10.1586	10.1586

Table 3.1: Minimized distances obtained by DMA and APM, $density = 1, rc = 0.01$

n	DMA, $\beta = 1$	APM
10	3.2030	3.2030
50	7.3449	7.3449
100	10.4252	10.4252

Table 3.2: Minimized distances obtained by DMA and APM, $density = 1, rc = 0.2$

Table 3.1 to 3.4 show that both algorithms converge to the same optimal point for a given symmetric matrix A. For all cases, the minimized distance obtained by both DMA

n	DMA, $\beta = 1$	APM
10	3.3032	3.3032
50	7.4549	7.4549
100	10.7306	10.7306

Table 3.3: Minimized distances obtained by DMA and APM, $density = .5, rc = 0.2$

n	DMA, $\beta = 1$	APM
10	3.4050	3.4050
50	7.7511	7.7511
100	10.9304	10.9303

Table 3.4: Minimized distances obtained by DMA and APM, $density = .5, rc = 0.4$

and APM are exactly identical. Therefore, we conclude that DMA is able to converge to the nearest correlation matrix, with β value equal to 1.

3.2 Efficiency comparison of DMA and APM

Results in Section 3.1 show that both DMA and APM converge to the same optimal point. Therefore, both solutions have the same accuracy. Thus in this section, we test how fast DMA and APM can converge to the optimal points, and we test that with different levels of accuracy. For each example, we run the numerical experiment 100 times based on APM Algorithm 2.1.1 and DMA Algorithm 2.2.3. Then we record the average numbers of iteration and CPU times of both algorithms.

In this numerical experiment, we set the tolerance (tol, i.e the accuracy of solution) to start from 1×10^{-8} up to 1×10^{-12} , and the maximum numbers of iteration (maxit) is 100. For the simplicity, we set the weighted matrix W to be the unit diagonal matrix.

Accuracy	DMA, iteration	DMA, CPU time	APM, iterations	APM, CPU time
1×10^{-8}	2.12	3.9503e-004	4.97	5.4747e-004
1×10^{-9}	2	3.7062e-004	4.56	4.9418e-004
1×10^{-10}	2.04	3.7836e-004	4.65	5.0547e-004
1×10^{-11}	2.04	3.7859e-004	4.79	5.2206e-004
1×10^{-12}	2.12	3.8943e-004	5.14	5.6112e-004

Table 3.5: Efficiency of DMA and APM with different accuracies, $n = 10, density = 1, rc = 0.4$

From Table 3.5 to 3.8, we can see that as the accuracy of solution required increases, APM uses more iterations and cpu time to converge to the optimal point, but the computational

Accuracy	DMA, iteration	DMA, CPU time	APM, iterations	APM, CPU time
1×10^{-8}	2.10	1.9673	5.23	2.0011
1×10^{-9}	2.23	2.0825	5.66	2.2087
1×10^{-10}	2.20	2.1081	5.63	2.2963
1×10^{-11}	2.22	2.3248	5.60	2.4869
1×10^{-12}	2.20	2.4258	5.84	2.4801

Table 3.6: Efficiency of DMA and APM with different accuracies, $n = 50, density = 1, rc = 0.4$

Accuracy	DMA, iteration	DMA, CPU time	APM, iterations	APM, CPU time
1×10^{-8}	3.83	0.0399	10.34	0.0489
1×10^{-9}	3.99	0.0427	11.00	0.0584
1×10^{-10}	4.05	0.0447	12.99	0.0678
1×10^{-11}	3.87	0.0414	13.11	0.0624
1×10^{-12}	3.88	0.0406	14.34	0.0692

Table 3.7: Efficiency of DMA and APM with different accuracies, $n = 100, density = .5, rc = 0.01$

Accuracy	DMA, iteration	DMA, CPU time	APM, iterations	APM, CPU time
1×10^{-8}	2	1.3400	5.21	1.3437
1×10^{-9}	2.86	1.86	7.87	2.3025
1×10^{-10}	3.53	2.26	9.86	3.0494
1×10^{-11}	2.90	2.021	8.83	2.8655
1×10^{-12}	3.60	2.28	12.10	3.6029

Table 3.8: Efficiency of DMA and APM with different accuracies, $n = 500, density = .4, rc = 0.01$

cost of DMA remains in the same level. Thus, when the accuracy of solution increases, the computational cost saved by DMA is more significant, especially when the original matrix A is having large size and ill-conditioned. Also we observe that DMA only take few iteration steps to find the optimal solution. This is because of the free movements of points in DMA. The optimal point obtained in one iteration during the DMA searching process do not need to be in the constraint sets, so it can move to the optimal solution following a short path. However, even though DMA can take much fewer iteration steps to find the optimal solution, the CPU time reduced by DMA is not very significant. The reason is that DMA need more time to find the optimal solution for one iteration. Even when β value 1, with which the direction equation 2.8 is in the simplest form 3.2, it still need to project three times and also involve few operations between those projections to finish one iteration, but APM only takes two projections and there is no any addition or deduction between those projections in one iteration. Therefore, we see that even though the difference of iteration steps between DMA and APM is large, the cpu times taken by both algorithms does not have the same big gap as iteration steps.

Actually, DMA is more popular to be applied on solving the nonconvex problems, especially when we are seeking global optimal solutions. For those problems, APM is often trapped by the local minimum and fail to converge to the global solution, but DMA is able to escape those traps. We use a simple example to illustrate the advantage of using DMA to solve nonconvex problems in Chapter 4.

Chapter 4

Global Convergence of DMA for Nonconvex Problems

We have shown that DMA has a higher performance than APM when we look for a high accurate solution to the NCM problem, which is a particular example of convex problems. However, The ability of finding global optimal solutions to nonconvex problems is a even bigger improvement provided by DMA.

It is shown that for nonconvex problems, original APM, which is defined in (2.1), can only achieve a local convergence at a linear rate [14]. Because of the existence of local minimums between nonconvex constraints, APM may move the point from one constraint set to another, and back to the starting point of the current iteration at the local minimum. This phenomenon is called *stagnation* [9]. (e.g. APM searching process stagnates at point X_2 in Figure 4.1.) Stagnation occurs very often when we apply APM to nonconvex problems and thus, APM generally fail to converge to global optimal solutions. (e.g. In Figure 4.1, APM starts from A_2 stagnates at the local minimized point X_2, and fail to find the global optimal solution.)

Unlike APM, DMA can avoid the stagnation if we set β value to be 1 [22]. (Numerical experiments show that if β value is too small, DMA may be trapped and fail to converge to the global optimal point [9, Chapter 8].) The reason is that every iteration of DMA moves the point in a specific direction, rather than projecting the point in one constraint onto the other one. The movement of the points is no longer confined by constraints. Thus, DMA has a bigger searching space, which makes it easily escape from the traps [22]. Also, because of the free movement of the point, it may converge to the optimal points following a short cut to reduce the number of iterations. For a same original matrix A, DMA may take much less iterations to reach the optimal solution than APM. (Numerical experiments in Section 3.2 show that DMA only take few iteration steps to reach the solution whereas APM takes much more iterations to find the same point.) We use a example in 2D graph to illustrate the different abilities of finding solutions to nonconvex problems between DMA and APM.

This example is a protein structure prediction problem, which is described in the paper [22]. The setting of this problem is to find the atom configuration which satisfies both constraints G and E simultaneously, Where G and E are the constraint sets. G is the space of atomic configuration that have a valid peptide geometry. This space contains all information about the atoms and bond connecting them. E is the space of atomic configurations whose non-bonded energy is less than a predefined target energy, E_0 [22]. Thus, G is a nonconvex set and E is affine space. The projection onto G is has a valid peptide geometry constraint, but the non-bouded energy may be larger than the predefined target energy, and the projection onto E has a non-bond energy less than E_0 but the configuration might not be peptide geometry [22]. The functions about these two constraints are defined in the appendix of [22].

The APM for this problem is defined similarly to (2.1) except the constraint sets [22]:

$$X_k \leftarrow P_G(P_E(X_{k-1})). \tag{4.1}$$

The iteration process stops at the point X^* such that $P_G(X^*) = P_E(X^*)$.

For DMA, the β value is chosen to be 1 and the formulas are defined as [22]:

$$X_k = X_{k-1} + \overrightarrow{\mathbf{d}}, \tag{4.2}$$

where

$$\overrightarrow{\mathbf{d}} = P_E(2P_G(X_{k-1}) - X_{k-1}) - P_G(X_{k-1}). \tag{4.3}$$

The solution is obtained at X^* when $\overrightarrow{\mathbf{d}} = 0$, or $P_E(2P_G(X^*) - X^*) = P_G(X^*)$ [22].

Figure 4.1 shows how DMA and APM searching iterations move. In Figure 4.1, constraint set G is indicated by two red circles and E is the space located on the left side of the blue line. (Thus, a point on the blue line indicated the precondition target energy E_0.) The grey dashed line shows the location of the intersection of constraints G and E, so the global solution to this protein structure prediction problem is any point on this dashed line. We have two starting points A_1 and A_2, and for each starting point, we apply both algorithms to find the optimal points. APM is shown by the lighter grey path and the optimal point of each APM iteration is colored by yellow; DMA is following the darker grey path, and the optimal point of each DMA iteration is in green.

From Figure 4.1 we can see that the APM searching iterations are confined by the constraints. Each APM iteration is moving in the direction to one of constraints and the optimal solution in every iteration has to be located in either of the constraints. Conversely, DMA moves the point directly towards to the intersection immediately at the starting point, and we can see that instead of being restricted in the small space between two constraints G and E, DMA can move the point anywhere in the entire space, and the optimal point obtained in each DMA iteration does not need to be in the constraint sets anymore. This property gives DMA the ability to find the shortest path to the solution. Therefore, even though both algorithms achieve the final solution starting from A_1, DMA takes fewer iteration steps than APM. For staring point A_2, APM fails to converge to the global optimal point by stagnating at the point X_2, where the distance between constraint G and E is locally minimized. DMA escaped from this local trap by moving the point into a bigger space and eventually find the actual solution [22]. Also, note that at X_2, since the stopping condition of APM is not satisfied (i.e, $P_G(X_2) \neq P_E(X_2)$), APM keeps doing the searching iteration perpetually.

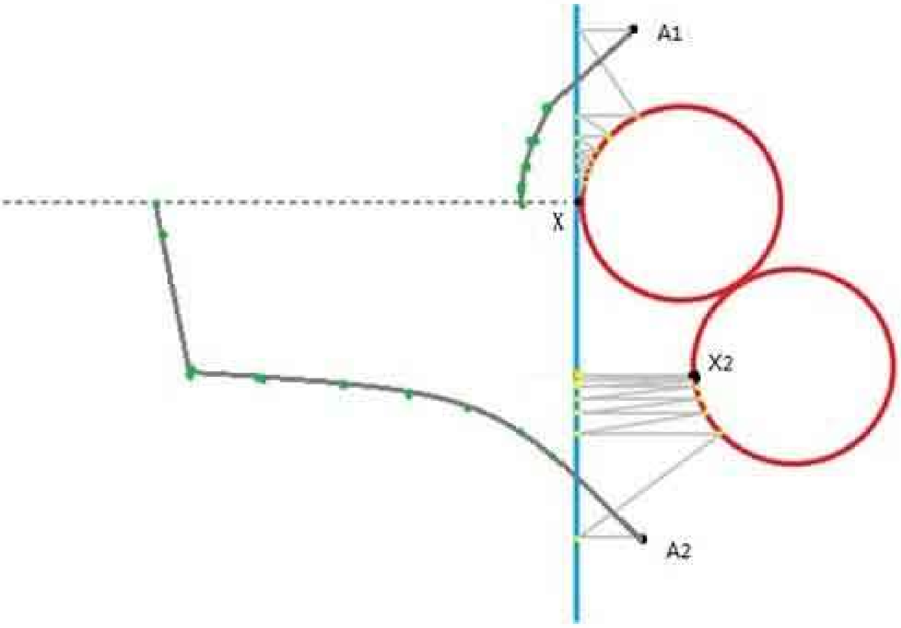

Figure 4.1: Protein structure predicted in the intersection of E and G

In Figure 4.1, we see that if APM starts from A_1, it may still converge to the global optimal solution X . However, we know that in general, the atomic structure of a protein is not as simple as shown in Figure 4.1. If we add another red circle on top of constraint G, (the position of this circle is shown in Figure 4.2,) such that there are two local minimums (X_1 and X_2) between constraint G and E, then we can see from Figure 4.2 that APM fails to find the global optimal solution from both starting points. APM searching process starting from A_1 stagnates at the local minimum X_1, and it stagnates at X_2 if starts from A_2. However, DMA can always converge to the point in the intersection of both constraints, either starting from A_1 or A_2. In reality, the atomic structure is much more complicated than what we draw in Figure 4.1 and 4.2. Several pictures can be seen in [22, Figure 4 and 5]. In fact there are many local minimums between constraint G and E. Those local traps can force APM to stagnate and fail to find the global optimal solution, but DMA can avoid them and reach the final solution. If we want to find a local minimum, APM is a useful algorithm to be applied. But if we are seeking the global optimal solution, DMA is a better algorithm to choose. Also, the numerical results in [22, Chapter 4] show that DMA is a very efficient algorithm for solving this protein structure prediction problems.

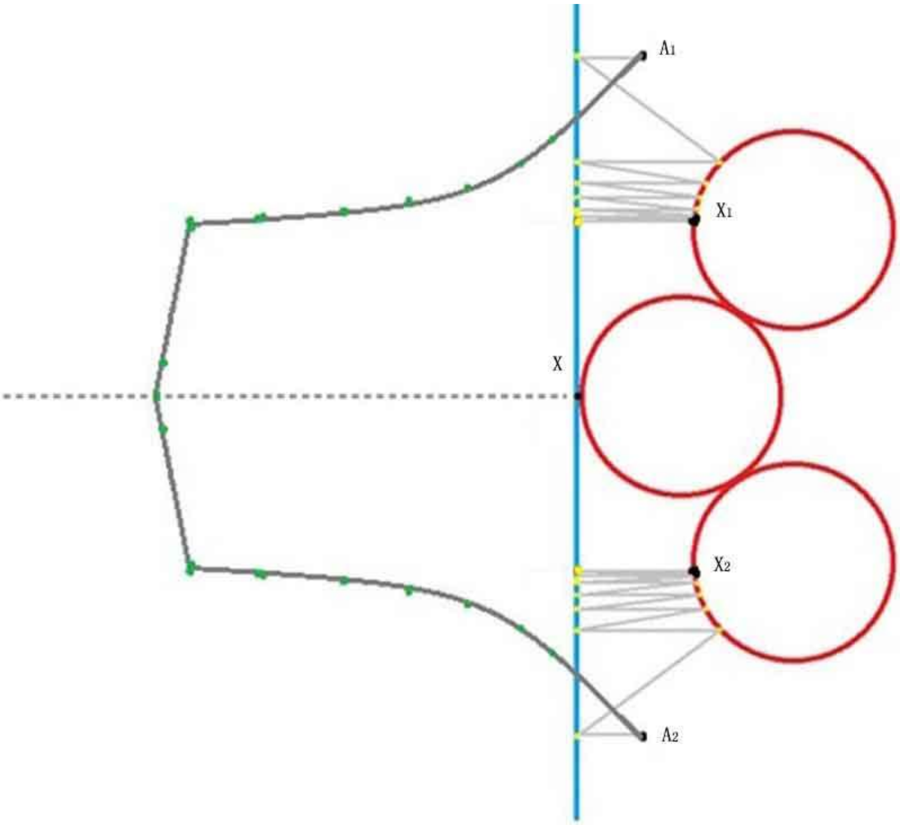

Figure 4.2: Protein structure predicted in the intersection of E and G

Chapter 5

Conclusion

In this thesis we introduce a new search algorithm called difference map algorithm (DMA) to solve the nearest correlation matrix problem. We find out that DMA may converge to different points in the intersection with different β values. We use numerical test to find the β value that can let DMA converge to the optimal solution, which is the nearest correlation matrix. Numerical experiments show that DMA is more efficient to find the optimal solution than APM when the high accuracy of optimal solution is required. Also when the original matrix A of the NCM problem is having large size and ill-conditioned, DMA presents a higher performance on finding the optimal solution than APM. Although, comparing to nonconvex problems, the improvement provided by DMA for convex problems is not very significant. For nonconvex problems, DMA can easily avoid the stagnation and find the global optimal points, whereas APM only can achieve the local minimized point; But for convex problems, the best solution obtained by DMA is the same as the APM, (both can find the global optimal solution,) and even though it can achieve the optimal solution in few iteration steps, the total CPU time saved by DMA is not very much because DMA needs more time to finish one iteration. However, we can still apply DMA to find the optimal solution to convex problems. especially when we are seeking the highly accurate solutions. We hope that this new algorithm can help us to minimized the errors when solving convex problems in practice.

Chapter 6

Appendix

Figures from 6.1 to 6.4 show that given a symmetric matrix A, DMA may converge to different optimal points in the intersection of the constraints if the β values are different. Only one of the points are the optimal solution (i.e, nearest correlation matrix) to A. Thus ,we need to determine the β value that give us the minimized one among all distances.

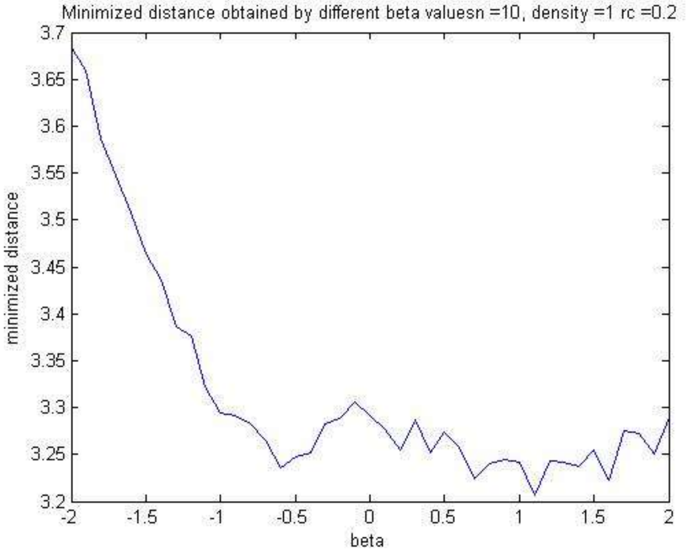

Figure 6.1: minimized distances obtained with different β, $n = 10$, $density = 1$, $rc = .2$

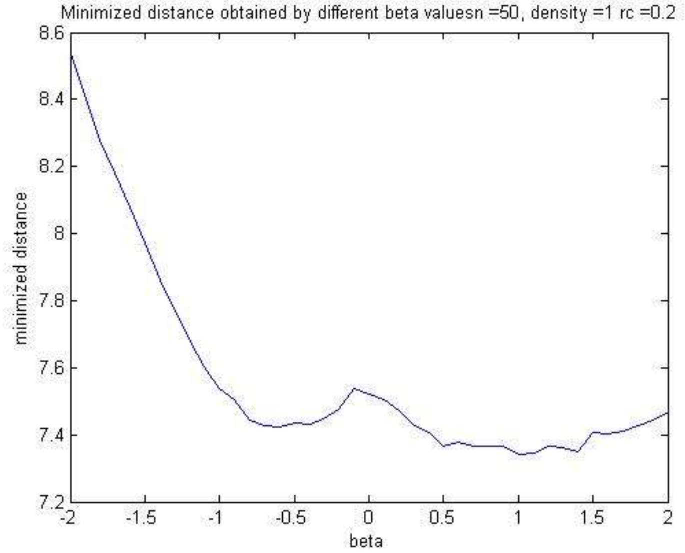

Figure 6.2: minimized distances obtained with different β, $n = 50$, $density = 1$, $rc = .2$

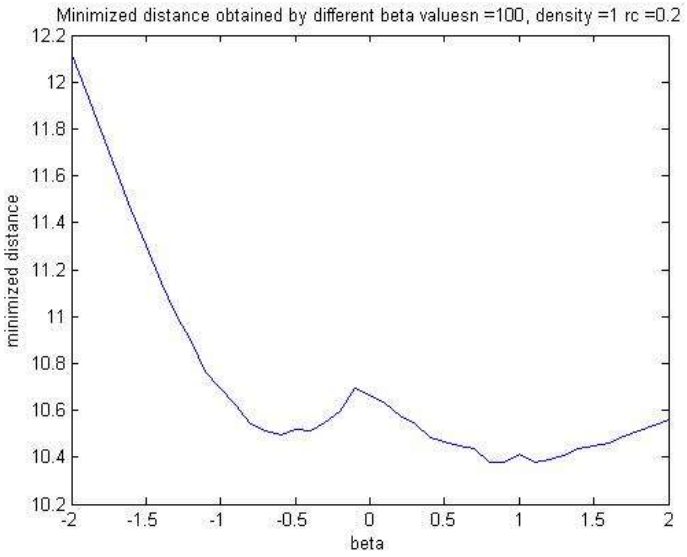

Figure 6.3: minimized distances obtained with different β, $n = 100$, $density = 1$, $rc = .2$

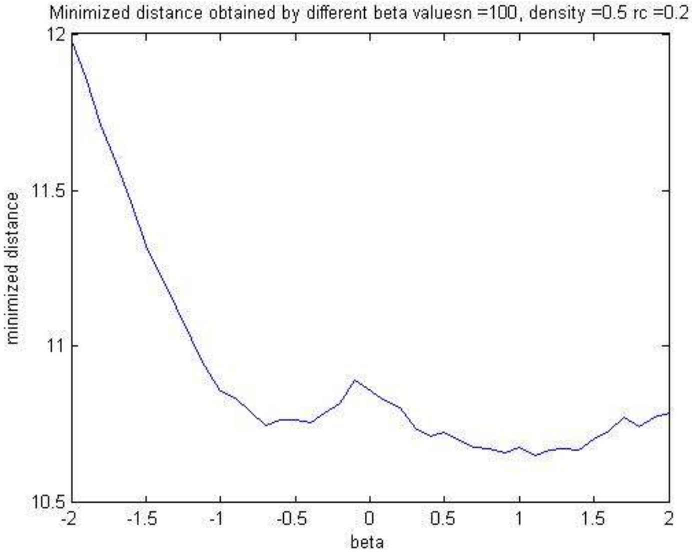

Figure 6.4: minimized distances obtained with different β, $n = 100$, $density = .5$, $rc = .2$

Bibliography

[1] M. Anjos, N. Krislock, M. Takouda, and H. Wolkowicz, *A semidefinite programming approach for the closest correlation matrix problem*, Tech. Report in progress, University of Waterloo, Waterloo, Ontario, 2009. 3

[2] N. Aronszajn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc., 68 (1950), pp. 337–404. 7

[3] H. Bauschke and J. Borwein, *On the convergence of von Neumann's alternating projection algorithm for two sets*, Set-Valued Anal., 1 (1993), pp. 185–212. 5

[4] J. Boyle and R. Dykstra, *A method of finding projections onto the intersection convex sets in Hilbert space*, Lecture Notes in Statistics, 37 (1986), pp. 28–47. 6

[5] F. Deutsch, *The method of alternating orthogonal projections*, in Approximation theory, spline functions and applications (Maratea, 1991), vol. 356 of NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., Kluwer Acad. Publ., Dordrecht, 1992, pp. 105–121. 5

[6] ——, *Best approximation in inner product spaces*, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, 7, Springer-Verlag, New York, 2001. 6

[7] F. Deutsch and H. Hundal, *The rate of convergence for the method of alternating projections. II*, J. Math. Anal. Appl., 205 (1997), pp. 381–405. 3, 6

[8] R. L. Dykstra, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 837–842. 5, 6

[9] V. Elser, *Phase retrieval by iterated projections*, ArXiv Mathematics e-prints, (2001). 7, 18

[10] B. Grone, C. Johnson, E. Marques de Sa, and H. Wolkowicz, *Positive definite completions of partial Hermitian matrices*, Linear Algebra Appl., 58 (1984), pp. 109–124. 1

[11] S.-P. Han, *A successive projection method*, Math. Programming, 40 (1988), pp. 1–14. 6

[12] N. Higham, *Computing the nearest correlation matrix—A problem from finance*, IMA J. of Numerical Analysis, 22 (2002), pp. 329–343. www.maths.man.ac.uk/ nareports/narep369.ps.gz. viii, 2, 3, 5, 6, 8, 9

[13] N. Krislock, F. Rendl, and H. Wolkowicz, *Noisy sensor network localization using semidefinite representations and facial reduction*, Tech. Report CORR 2010-01, University of Waterloo, Waterloo, Ontario, 2010. 3

[14] A. S. Lewis and J. Malick, *Alternating projections on manifolds*, Math. Oper. Res., 33 (2008), pp. 216–234. 18

[15] C. Lucas, *Computing nearest covariance and correlation matrices*, thesis, University of Manchester, 2005. 1

[16] H. M. Markowitz, *Portfolio selection: Efficient diversification of investments*, Cowles Foundation for Research in Economics at Yale University, Monograph 16, John Wiley & Sons Inc., New York, 1959. 1

[17] ——, *Mean-variance analysis in portfolio choice and capital markets*, Basil Blackwell, Oxford, 1987. 1

[18] ——, *Harry Markowitz. Selected works*, vol. 1 of Nobel Laureate Series, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2008. 1

[19] MATLAB, sprandsym*, sparse symmetric random matrix*, in Documentation, The MathWorks, 2010. 11, 12

[20] H. Qi and D. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 360–385 (electronic). 3

[21] ——, *An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem*, IMA J Numer Anal, (2010), p. drp031. 9

[22] I. C. Rankenburg and V. Elser, *Protein structure prediction by an iterative search method*, ArXiv e-prints, (2007). 1, 2, 18, 19, 20

[23] W. Sharpe, *The Sharpe ratio*, Journal of Portfolio Management, Fall (1994), pp. 49–58. 1

[24] J. von Neumann, *Functional Operators. II. The Geometry of Orthogonal Spaces*, Annals of Mathematics Studies, no. 22, Princeton University Press, Princeton, N. J., 1950. 5

Index