Low-Rank Completion and Recovery of Correlation Matrices

Chetan K. Ramlall

A dissertation submitted to the Faculty of Commerce, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

August 23, 2019

MPhil in Mathematical Finance,
University of Cape Town.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted as partial fulfillment of the requirements for the Degree of Master of Philosophy in Mathematical Finance, University of Cape Town. It has not been previously submitted for any examination or degree in any other University.

Signed by candidate

_____

Chetan K. Ramlall
August 23, 2019

Abstract

In the pursuit of efficient methods of dimension reduction for multi-factor correlation systems and for sparsely populated and partially observed matrices, the problem of matrix completion within a low-rank framework is of particular significance. This dissertation presents the methods of spectral completion and convex relaxation, which have been successfully applied to the particular problem of low-rank completion and recovery of valid correlation matrices. Numerical testing was performed on the classical exponential and noisy Toeplitz parametrisations and, in addition, to real datasets comprising of FX rates and stock price data. In almost all instances, the method of convex relaxation performed better than spectral methods and achieved the closest and best-fitted low-rank approximations to the true, optimal low-rank matrices (for some rank-n). Furthermore, a dependence was found to exist on which correlation pairs were used as inputs, with the accuracy of the approximations being, in general, directly proportional to the number of input correlations provided to the algorithms.

Acknowledgements

I would like to thank my supervisor, A/Prof. Thomas McWalter and co-supervisor, James Taylor, for their invaluable input, guidance and expertise throughout this dissertation. In addition, I express my thanks to Prof. David Taylor and the AIFMRM team, for affording me the opportunity to pursue this rewarding degree and for the continued guidance and commitment of all the lecturing staff. My sincerest thanks also goes to STANLIB and the National Research Foundation (NRF), for the generous funding and resources, without which this journey would not have been possible. Finally, I extend my heartfelt thanks to my parents, grandparents, extended family and friends for their ongoing love and support in all my endeavours, especially during this challenging year, for which I will be forever grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The problem of low-rank matrix completion and recovery is one that arises in many areas of the sciences, economics and finance. For sparsely populated data sets, and large matrices with partial observations, the process of estimating and filling in the missing entries (in the most accurate sense) is known as matrix completion. This is typical of correlation matrices (Pham, 2013), which are commonly observed in asset and interest rate modelling problems. As more factors are added to a system, these correlation matrices rapidly become larger and more complex, with more unknown correlations requiring determination. There are also corresponding increases in both computational and modelling complexities, thereby promoting the need for low-rank approximations, in pursuit of the ultimate goal of dimension reduction.

The applications of low-rank correlation matrix approximations within finance are quite varied, a common example being the pricing of basket options, where the correlations between a number of stocks are being considered. It can also be applied to multi-factor interest rate modelling (e.g. the LIBOR Market Model) where high-dimensional correlation matrices are commonplace when modelling large sets of forward rates. Even in multi-factor simulations of FX rates, low-rank approximations find significant relevance.

Interestingly, the literature points to there being two distinctive fields in this area of research. There has been significant work done on the correlation matrix nearness problem, which deals with finding the optimal low-rank approximation to a correlation matrix. On the other hand, the problem of low-rank matrix completion (for arbitrary matrices) has garnered substantial attention over the last decade, with the advent of machine learning, compressed sensing (for signal reconstruction), recommender systems and improved optimisation techniques. However, the area of low-rank completion with respect to correlation matrices in particular, has not been directly addressed in the literature nor analysed in any significant detail. It is with this in mind, that we approach this problem in earnest.

Typically, in multi-factor modelling, Brownian motion is used in the modelling of assets, and Hull-White processes for interest rates. This results in a set of correlated stochastic differential equations (SDEs), which have Brownian motion as an underlying driver. As an example, consider currency trades between two banks. In a challenging situation, assume that there are 10 currencies, with 9 corresponding currency pairs (with respect to one of the currencies that is taken as a base). If we require a model of the exposure between one bank and another, the model would need to include all 10 interest rates for each of the countries, as well as all 9 cur-

rency pairs, yielding a total of 19 factors. This results in 171 possible correlations, an amount that would rapidly increase as more factors are added to the system. Naturally, estimating these large sets of correlations becomes increasingly difficult.

This leads to the following research question: An individual seeks to model correlations between multiple factors; what is the minimum number of correlation pairs that need to be provided as input and which input correlation pairs are the most important, to most reliably recover the closest low-rank approximation of the "true", low-rank matrix?

This dissertation begins by highlighting some of the key concepts that are central to this problem, including the relevant error metrics used for analysis. Applications of low-rank approximations and completion are then discussed and a brief review of the Netflix Prize is provided (Koren, 2009), which popularised the low-rank completion problem in recent times. This is followed by a review of some of the important techniques of matrix decomposition, such as principal component analysis (PCA) and the singular-value decomposition (SVD). The next section introduces the correlation matrix nearness problem, which aims to determine the optimal low-rank approximation to a correlation matrix. The preeminent techniques of modified PCA, angles parametrisation and majorisation are presented herein. The final section of the literature review deals with general low-rank matrix completion; the theoretical formalism and underlying assumptions. The work of Candès and Recht (2009) serves as the basis for the theory, and important notions of permissible low-rank forms and matrix coherence are introduced. Two techniques of general low-rank completion, the first by convex relaxation and second by non-convex optimisation, are then motivated and presented.

Chapter 3 deals with the construction of real correlation matrices for the purposes of testing. The classical exponential form, as proposed by Brigo and Mercurio (2006), is described, as well as a noisy Toeplitz parametrisation, proposed by Hardin *et al.* (2013). In Chapter 4, the completion problem is addressed directly, and is considered from two perspectives. The first deals with preserving the original user-defined low-rank matrix, while the second deals with allowing the original user-defined matrix to be modified, in pursuit of greater overall accuracy in the low-rank approximation. The first proposed technique, which we have termed spectral completion, is presented, which consists of two independent methods based on correlations and covariances respectively. The second method of completion by convex relaxation is then described and the testing procedure expounded in detail.

Extensive numerical results are presented in Chapter 5, for both the classical and noisy Toeplitz correlation parametrisations, and a comprehensive comparison of the performance of each of the different completion methods was undertaken. Testing was performed for both fully and under-determined systems, where fewer correlation inputs were provided than unknowns in the system. Real FX rates data, as well as stock price data for the Top 10 SA stocks, was obtained, and real correlation matrices were generated. The completion algorithms were then tested on these real matrices and the performance of each was compared. The dissertation concludes with a general discussion of results, limitations and extensions to the work, as well as a summary of the main conclusions drawn.

Chapter 2

Literature Review

2.1 Key Concepts

Before delving directly into the main body of work, a few important concepts are reviewed, which are central to this dissertation. They are intended to highlight, with a degree of brevity, some of the fundamental theory and to aid overall exposition and understanding.

2.1.1 Basic Matrix Theory

Definition 2.1 (Matrix Rank)**.** The rank of a matrix is defined as the number of linearly independent columns (or rows) it contains i.e. it is the dimension of the vector space spanned by its columns (or rows). Consider the matrix below:

$$\mathbf{M} = \begin{pmatrix} 1 & 3 & 7 \\ 2 & 6 & 4 \\ 3 & 9 & 5 \end{pmatrix}$$

In this case $\text{rank}(\mathbf{M}) = 2$ as columns 1 and 2 are linearly dependent.

Definition 2.2 (Eigendecomposition of a Matrix)**.** The eigendecomposition (or spectral decomposition) of a matrix is a factorisation whereby the matrix is represented canonically in terms of its eigenvalues and eigenvectors. For a full-rank, square $N \times N$ matrix $\mathbf{M}$, it may be factorised as

$$\mathbf{M} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}, \tag{2.1}$$

where $\mathbf{Q}$ is a $N \times N$ matrix whose columns are eigenvectors and $\mathbf{\Lambda}$ is an $N \times N$ diagonal matrix whose diagonal elements are the corresponding eigenvalues. In the special case of real, symmetric matrices, (2.1) may be re-formulated as:

$$\mathbf{M} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\top}, \tag{2.2}$$

where $\mathbf{Q}$ is now an orthogonal eigenvector matrix, and $\mathbf{\Lambda}$ is given as above.

2.1.2 Important Matrix Norms

Definition 2.3 (Frobenius Norm)**.** The Frobenius norm of a matrix **A** (also known as the Euclidean norm) is defined as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}.$$

Alternatively, it can be expressed as

$$\|\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{A}\mathbf{A}^\top),$$

where $\mathrm{tr}(\mathbf{A})$ is the trace of matrix **A**. This is an important and useful matrix norm, and is extensively used in the measurement of error in analysis.

Definition 2.4 (Nuclear Norm)**.** The nuclear norm of a matrix **M** is defined as the sum of its singular values σ_i (see Section 2.3.3 for the Singular-Value Decomposition). Mathematically, this is expressed as

$$\|\mathbf{M}\|_* = \sum_{i=1}^{r}\sigma_i, \tag{2.3}$$

where r is the rank of **M**.

2.1.3 Error Metrics

There are two primary error metrics that will be used extensively for analysis throughout this dissertation. These have been termed the *matrix norm error (MNE)* and *eigenvalue norm error (ENE)* respectively, and are defined as follows:

Definition 2.5 (Matrix Norm Error)**.** The MNE is defined as the Frobenius norm between two matrices, i.e.

$$X_F = \|\mathbf{A} - \mathbf{B}\|_F,$$

where matrices **A** and **B** are of the same size.

Definition 2.6 (Eigenvalue Norm Error)**.** The ENE is defined as the Frobenius norm between the eigenvalue matrices of two matrices **A** and **B**, following an eigendecomposition. In other words,

$$X_{EV} = \|\mathbf{\Lambda_A} - \mathbf{\Lambda_B}\|_F,$$

where matrices $\mathbf{\Lambda_A}$ and $\mathbf{\Lambda_B}$ are the diagonal, ordered eigenvalue matrices of **A** and **B** respectively.

2.2 Applications of Low-Rank Approximation and Completion

As alluded to in Chapter 1, the applications of low-rank approximations within the context of finance are very important. Low-rank correlation structures in particular, are extensively used in the modelling of large sets of assets and interest rates, especially multi-rate dependent interest rate derivatives such as Bermudan swaptions (Pietersz and Groenen, 2004). Libor market models (LMMs) are another common example, where high-dimensional correlation structures are often required.

However, this problem of general low-rank matrix completion and recovery extends to many areas outside the sphere of finance. Chi (2018) notes the proliferation of research done in areas of application ranging from computer vision, compressed sensing, optimisation and machine learning. Candès and Recht (2009) provide the following illustrative example, which involves triangulation from incomplete data. Suppose we aim to determine the low-dimensional geometry of the locations of different objects, with only partial information given about the distances between them. Suppose there exists a network of wireless sensors scattered randomly in the area, where distance estimates can be constructed based on the signal strength of neighbouring sensors. A partially observed distance matrix can now be formed from these noisy distance estimates. The true, complete distance matrix can then be estimated. If the sensors are located in a plane, this estimated matrix will have a rank of two, and if they are arranged in three-dimensional space, it will be of rank three. In essence, we only require observations of a few nodal distances within this network, to have sufficient information to recover the full distance matrix (of some low-rank).

2.2.1 The Netflix Prize

Interestingly, this very problem and associated techniques found an adopter in a rather unexpected area. The problem of low-rank completion and recovery was popularised in recent times by the so-called Netflix Prize (Koren, 2009). Netflix, the U.S. based media streaming service-provider, offered a prize of a $1million to anyone who could improve their recommendation algorithm (known as CineMatch) by 10% or more. The competition was launched in October 2006, and by June 2007, over 20000 teams from 150 countries were competing. Essentially, the more data Netflix gathers, the better the algorithm becomes at predicting viewing preferences. It utilises the method of collaborative filtering, which uses multiple input sources of data (in this case the ratings by other viewers), to filter information (Koren, 2009).

The dataset consisted of 100 million ratings, 18000 movies and approximately 480000 anonymous Netflix users. On average, each movie was rated by 5000 users and each user rated over 200 movies. Clearly, this was a very sparse dataset, with only about 1% of the total number of possible ratings known. Furthermore, the data was skewed, as different users rated different numbers of movies. This presented the competitors with the challenge of dealing with a very large, skewed and sparse dataset (Koren, 2009). The aim was to find a reduction in the root-mean-squared error (RMSE), which is a measure of the difference between model-

predicted values and those observed. The eventual winning solution, by team *Bellkor's Pragmatic Chaos*, consisted of an averaging of over 800 different algorithms. These were based on an extensive variety of techniques, the primary among them being neighbourhood model-based algorithms with collaborative filtering, with associated techniques of convex and non-convex optimisation. Neighbourhood models were found to be useful in matrix completion, as they dealt with estimating and explaining values based on their proximity to other data points within the low-rank structure (Koren, 2009).

2.3 Matrix Decomposition Techniques

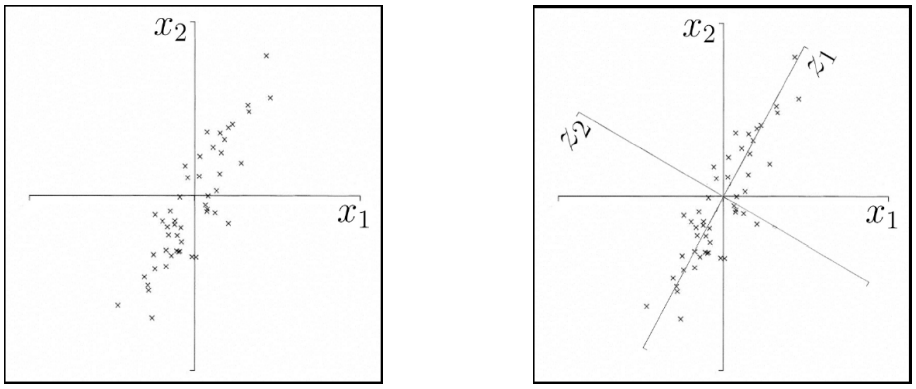
The following section deals with a few important techniques involved in the decomposition of matrices, as part of the rank-reduction problem. It provides an overview of some of the key mathematical ideas involved, together with the theoretical formalism.

2.3.1 Principal Component Analysis

Principal component analysis (PCA) is a very powerful technique used in the compression and classification of data. It decomposes a dataset and extracts the so-called principal components, which represent most of the information in the sample. Information in this sense, refers to the amount of variation in the sample, which is given by the correlations between the original set of variables. PCA finds a new orthogonal basis which represents the majority of the information of the original sample. These principal components (PCs) are themselves uncorrelated, whereby they each account for a portion of the variation in the original dataset and are ordered by the fraction of the total information each retains (Caveny and Murray, 1995). It should be noted that the number of PCs extracted may be as large as the number of original variables, and that PCs can keep being produced until all of the variability of the original dataset has been accounted for. In practice, however, the first few PCs contain the majority of the information (Karamizadeh *et al.*, 2013) of the sample, and it is often up to the user to decide on the number of practically useful PCs to extract. In general, the smallest number of PCs should be chosen, which account for the largest amount of variability.

Consider the plots in Figure 2.1, which represent the spread of some arbitrary data in $x_1 - x_2$ space.

As can be seen, the distribution of this test data is elliptical in shape. Consider the line drawn through the major axis of the ellipse – this is the dimension along which most of the variability lies. We define this line z_1 as the first PC, which represents the minimum distance fit of the data to the line (Caveny and Murray, 1995). The remaining variability of the data can be accounted for by a line drawn perpendicular to the first, z_2, which represents the second PC. It is the minimum distance fit of the data to the line in a plane orthogonal to the first PC. In essence, PCs can be thought of as a series of linear least-squares fits to a dataset, where each new PC is



(a) Data points in Cartesian space. **(b)** Data points with two PCs $Z_1$and Z_2.

Fig. 2.1: Data points with fitted principal components.

orthogonal to all preceding PCs. Practically, PCA involves decomposing a matrix into its eigenvalues and eigenvectors and finding the largest eigenvalues and their corresponding eigenvectors. The first eigenvalue recovers the most variance, the second eigenvalue recovers the second-most, and so on.

As part of the particular problem we are trying to address, which requires finding low-rank approximations to correlation matrices, a reduced-rank correlation matrix can be constructed by taking a selection of these eigenvalues and eigenvectors (to be discussed in Section 2.4.1). This lower-rank correlation matrix typically recovers the majority of the variance of the initial full-rank matrix.

2.3.2 The Moore-Penrose Inverse

Before exploring the idea of the Moore-Penrose inverse, the first notion to consider is that of the generalised inverse. From elementary linear algebra, the inverse of a matrix is only defined for square matrices with non-zero determinants (i.e. non-singular matrices). There is no notion of inverting singular matrices, nor those of rectangular form. However, in recent years, there has been a growing need in many areas of applied mathematics to develop "inverse-like" matrices for precisely those of the aforementioned types (singular and/or rectangular) (MacAusland, 2014). It was subsequently discovered that even for non-invertible matrices, there still exists either a left or right-sided inverse of that matrix. A matrix $\mathbf{A} \in \mathbb{C}^{n\times m}$ (where $\mathbb{C}$ is the set of complex numbers) is left-invertible (right-invertible) if there is a matrix $\mathbf{L}(\mathbf{R}) \in \mathbb{C}^{n\times m}$, such that

$$\mathbf{LA} = \mathbf{I}_n \qquad (\mathbf{AR} = \mathbf{I}_m),$$

where $\mathbf{I}_{n,m}$ is the identity matrix of corresponding size. It is the above property that lends to the definition of the generalised inverse, which has uses in areas such as statistics, eigendecomposition and inconsistent least-squares systems. In practice, however, the generalised inverse is not used, with preference being given to the so-called Moore-Penrose inverse. The Moore-Penrose inverse is part of a class of pseudoinverses (of which it is the most popular) that can be applied to either

under or over-determined systems. It is related to the least-squares operator and can be used to achieve a least-squares fit with data. As per MacAusland (2014), the following definition is given:

Definition 2.7 (Penrose Conditions)**.** If $\mathbf{A} \in \mathbb{C}^{n\times m}$, then there exists a unique $\mathbf{A}^+ \in \mathbb{C}^{n\times m}$ that satisfies the four Penrose conditions:

1. $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$
2. $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$
3. $\mathbf{A}^+\mathbf{A} = (\mathbf{A}^+\mathbf{A})^*$ (Hermitian)
4. $\mathbf{A}\mathbf{A}^+ = (\mathbf{A}\mathbf{A}^+)^*$ (Hermitian),

where $\mathbf{A}^*$ is the conjugate transpose of matrix $\mathbf{A}$ and $\mathbf{A}^+$ is the pseudoinverse. It should be noted that $\mathbf{A}^+ = \mathbf{A}^{-1}$ (i.e. the pseudoinverse simply reduces to the ordinary inverse), whenever $\mathbf{A}$ is non-singular.

It is in the solving of least-squares systems of the form $\mathbf{A}\vec{x} = \vec{b}$ that the pseudoinverse finds its primary use. When $\vec{b}$ is not in the column space of $\mathbf{A}$, no solutions exist for the system, but if $\vec{b}$ is in the column space of $\mathbf{A}$, then one or more solutions will exist (MacAusland, 2014).

Definition 2.8 (Least-Squares Solution)**.** A least-squares solution to a system is a vector $\vec{x}_0$ such that,

$$\|\mathbf{A}\vec{x}_0 - \vec{b}\| \leq \|\mathbf{A}\vec{x} - \vec{b}\|,$$

where $\|\mathbf{A}\vec{x}_0 - \vec{b}\|$ is defined as the residual vector $\vec{r}_0$. The unique least-squares solution is given when the vector $\vec{x}_0$ creates a minimum in the norm of the residual vector $\vec{r}_0$.

The following theorem is also of importance:

Theorem 2.9 (Approximation by Pseudoinverse)**.** *$\vec{x}_0 = A^+\vec{b}$ is the best approximate solution of $A\vec{x} = \vec{b}$.*

The utility of Theorem 2.9 lies in the fact that it can be used to determine whether the solution ($\mathbf{A}^+\vec{b}$) is a unique least-squares solution or the least-squares solution of minimum norm. Where this concept of the pseudoinverse may have relevance is in the case of low-rank completion by non-convex optimisation methods (as discussed in Section 2.5.4). Here, the problem may be formulated as a rank-constrained least-squares problem, as per (2.10).

2.3.3 Singular-Value Decomposition (SVD)

The singular-value decomposition (SVD) is a very popular and widely applied technique of matrix factorisation. In the 1970s, an efficient algorithm for its computation was developed, after which the SVD gained popularity. It is used widely in data analysis, can be generalised to higher dimensions and can be calculated for dense, structured or sparse matrices. The following two theorems, as presented in Martin and Porter (2012), provide the mathematical description of the SVD:

Theorem 2.10 (Singular-Value Decomposition). *Any matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ can be factored into a singular-value decomposition (SVD),*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}, \tag{2.4}$$

where $\boldsymbol{U} \in \mathbb{C}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{C}^{n\times n}$ are orthogonal matrices (i.e. $\boldsymbol{U}\boldsymbol{U}^{\top} = \boldsymbol{V}\boldsymbol{V}^{\top} = \boldsymbol{I}$) and $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ is rectangular with $r = \operatorname{rank}(\boldsymbol{A})$ leading non-negative, real diagonal entries. The p diagonal entries of $\boldsymbol{\Sigma}$ are usually denoted by σ_i for $i = 1, ..., p$, where $p = \min(m, n)$, and σ_i are called the singular values of $\boldsymbol{A}$. The singular values are the square roots of the non-zero eigenvalues of both $\boldsymbol{A}\boldsymbol{A}^{\top}$ and $\boldsymbol{A}^{\top}\boldsymbol{A}$, and they satisfy the property $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p$. (Geometrically, $\boldsymbol{U}$ and $\boldsymbol{V}^{\top}$ represent rotations, while $\boldsymbol{\Sigma}$ represents a stretch).

Note that (2.4) can also be written as a sum of rank-1 matrices,

$$\mathbf{A} = \sum_{i=1}^{r} \mathbf{u}_i \sigma_i \mathbf{v}_i^{\top}, \tag{2.5}$$

where σ_i is the ith singular value, and $\mathbf{u}_i$ and $\mathbf{v}_i$ are the ith columns of $\mathbf{U}$ and $\mathbf{V}$. This form is very useful in the rank-reduction problem, where we need to estimate $\mathbf{A}$ using a low-rank matrix.

Theorem 2.11 (Eckart-Young). *Let the SVD of $\boldsymbol{A}$ be given by (2.5). If $k < r = \operatorname{rank}(\boldsymbol{A})$ and $\boldsymbol{A}_k = \sum_{i=1}^{k} \boldsymbol{u}_i \sigma_i \boldsymbol{v}_i^{\top}$, then*

$$\min_{rank(\boldsymbol{B})=k} \|\boldsymbol{A} - \boldsymbol{B}\|_2 = \|\boldsymbol{A} - \boldsymbol{A}_k\|_2 = \sigma_{k+1}.$$

Theorem 2.11 represents one of the most useful features of the SVD, as it finds direct use in least-squares approximations and principal component analysis (PCA). The Eckart-Young theorem also proves to be very important in the minimisation of the error (norm) during the low-rank approximation of $\mathbf{A}$.

2.4 Optimal Low-Rank Approximations to a Correlation Matrix

In the area of dimension reduction, low-rank matrix approximations play an important role. The lower the rank of a system, the fewer factors drive it, which in turn leads to reduced computational and modelling complexities. As such, finding the optimal low-rank approximation to a matrix (and in particular correlation matrices) is therefore a very important problem, which has been addressed by a number of authors in the literature. Rebonato and Jäckel (1999), Zhang and Wu (2003) and Pietersz and Groenen (2004), have presented various techniques, each with particular advantages and limitations. Primary amongst these methods are modified principal component analysis (PCA), the method of angles parametrisation and the method of majorisation, which was proposed by Pietersz and Groenen (2004). Zhang and Wu (2003) also proposed a technique using Lagrange multipliers, however, this was not widely adopted. Consider the spectral decomposition of a correlation matrix $\mathbf{R} = \mathbf{X}\mathbf{X}^{\top}$, where $\mathbf{X}$ is an $n \times n$ matrix. If the ith row vector of $\mathbf{X}$

is denoted by $\mathbf{x}_i$, then the decomposition can be expressed as $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = r_{ij}$, where $\langle ., . \rangle$ represents the scalar product. Thus, the problem of finding the optimal low-rank matrix is generally specified as a minimisation problem. Pietersz and Groenen (2004) frame the objective in the following form:

Central Objective:
Find $\mathbf{X} \in \mathbb{R}^{n \times d}$ to minimise $f(\mathbf{X}) = \frac{1}{c} \sum_{i<j} w_{ij}(r_{ij} - \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^2$, subject to $\|\mathbf{x}_i\|_2 = 1$ where $i = 1, \ldots, n$ and w_{ij} are a set of element-wise weights.

Considering the simplest case, with all weights equal, the above formulation simplifies to $f(\mathbf{X}) := c^{-1} \|\mathbf{R} - \mathbf{X}\mathbf{X}^\top\|_F^2$. The constant c is used to scale the objective function f to make it independent of the problem dimension n, and is defined as $c := 4 \sum_{i<j} w_{ij}$ (Pietersz and Groenen, 2004).

2.4.1 Modified Principal Component Analysis (PCA)

Modified PCA is the first popular technique, which is applied by many practitioners in financial modelling (Pietersz and Groenen, 2004). It is a very simple and robust technique, which allows for efficient rank-reduction with a reasonable degree of accuracy (Rebonato and Jäckel, 1999). It is based on an eigendecomposition $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, as per (2.2). If the eigenvalues are arranged in descending order in the $\mathbf{\Lambda}$ matrix, then a low-rank decomposition with an approximated, close-to-the-original matrix is found by the following (Pietersz and Groenen, 2004):

$$\{\mathbf{X}_{\text{PCA}}\}_i = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}, \tag{2.6}$$

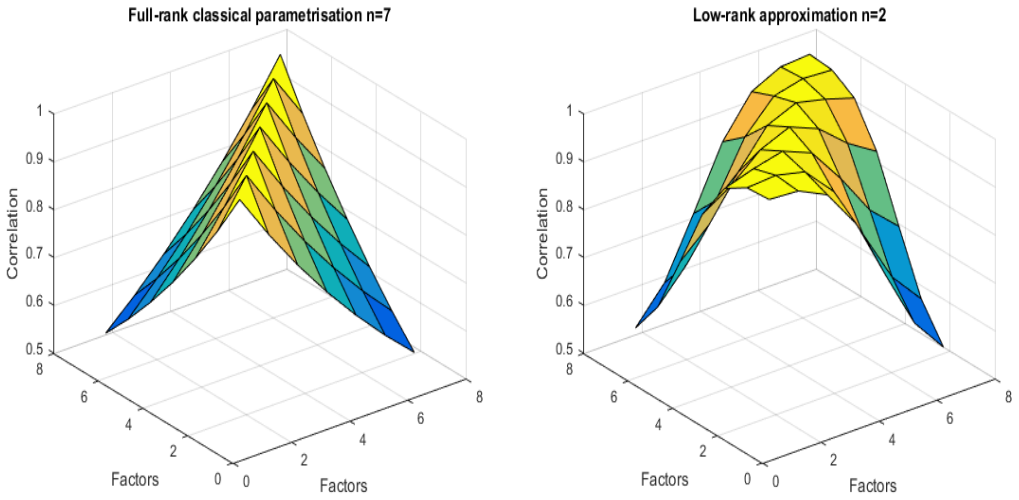
$$\mathbf{z} := \{\mathbf{Q}_d \mathbf{\Lambda}_d^{1/2}\}_i, i = 1, \ldots, n. \tag{2.7}$$

In the above formulation, $\{\mathbf{A}\}_i$ denotes the i^{th} row of a matrix $\mathbf{A}$, $\mathbf{Q}_d$ the first d columns of $\mathbf{Q}$, and $\mathbf{\Lambda}_d$ the principal d-degree sub-matrix of $\mathbf{\Lambda}$. The "modified" nature of this technique is due to the scaling in (2.6), whereas ordinary PCA stops at (2.7). This scaling ensures that the ones criterion is maintained along the main diagonal of the resulting correlation matrix. This technique has the advantage of being easy to implement, as it is essentially based on an eigendecomposition, and results in reasonably accurate approximations. One of the disadvantages, however, lies in its non-optimality. In general, there may exist decompositions $\mathbf{X}$, where the associated correlation matrix $\mathbf{X}\mathbf{X}^\top$ is closer than the PCA-approximated matrix $\mathbf{X}_{\text{PCA}}\mathbf{X}_{\text{PCA}}^\top$, to the original matrix $\mathbf{R}$ (Pietersz and Groenen, 2004). Naturally, as the magnitude of the omitted eigenvalues increases, the accuracy of the modified PCA approximation decreases.

Table 2.1 summarises the errors for rank-recovery by modified PCA, up to different ranks. Testing was performed for both the classical (with $\rho_\infty = 0$) and noisy Toeplitz (with $M = 3$) parametrisations (see Chapter 3), using 7×7 matrices. As can be seen, when the recovery is done to higher ranks, more information is retained in the system, resulting in a rapid decrease in error between the approximated and true, full-rank matrices. The improvement in the approximations are almost ten-fold, as the rank increases from two to six.

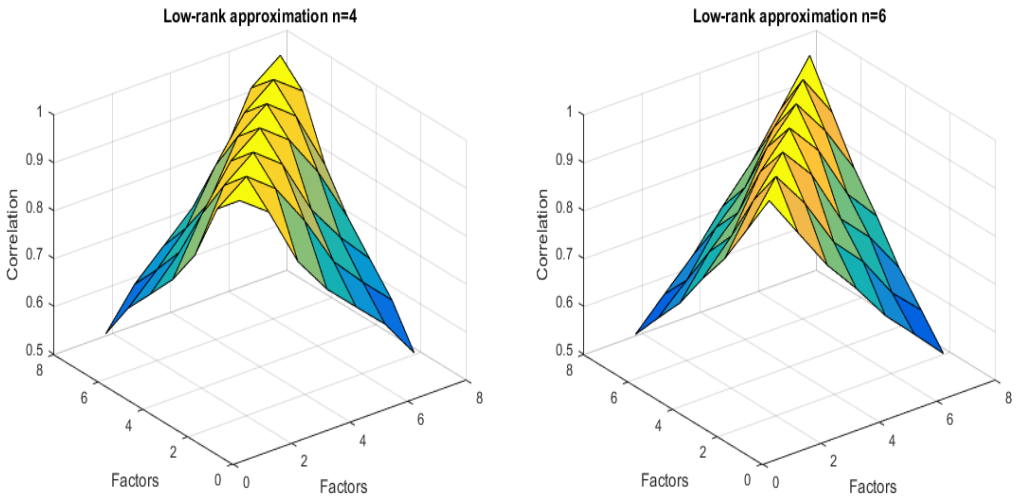
Tab. 2.1: Comparison of errors for recovery by modified PCA to different ranks.

	Classic		Noisy	
Rank (n)	X_F	X_{EV}	X_F	X_{EV}
2	0.59527	0.58819	0.59685	0.58778
4	0.20757	0.20255	0.15454	0.14657
6	0.07306	0.06964	0.03223	0.08885



(a) Full-rank with $n = 7$. **(b)** Low-rank with $n = 2$.

Fig. 2.2: Modified PCA applied to classical parametrisation.



(a) Low-rank with $n = 4$. **(b)** Low-rank with $n = 6$.

Fig. 2.3: Further low-rank approximations by modified PCA.

2.4.2 Angles Parametrisation

The angles parametrisation method, proposed by Rebonato and Jäckel (1999), is another prominent technique that has found relative success in the field of low-rank approximation. It is a technique based on a hypersphere decomposition and performs the approximations in terms of spherical coordinates defined by trigonometric sine and cosine functions. It is based on the well-known result that every $n \times n$ symmetric, positive, semi-definite matrix $\mathbf{R}$ may be decomposed as

$$\mathbf{M} = \mathbf{W}\mathbf{W}^\top,$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is also positive, semi-definite. Rebonato and Jäckel (1999) propose that for the construction of a valid correlation matrix $\hat{\mathbf{C}} = \mathbf{B}\mathbf{B}^\top$ that best approximates a target matrix $\mathbf{C}$, is to view the elements of the row vectors of matrix $\mathbf{B}$ as coordinates lying on a unit hypersphere. Denoting the elements of $\mathbf{B}$ by b_{ij}, the aim is to use the $n \times (n-1)$ angular coordinates θ_{ij} to obtain the $n \times n$ coordinates b_{ij}. Rebonato and Jäckel (1999) define these as follows:

$$b_{ij} = \cos\theta_{ik} \prod_{k=1}^{j-1} \sin\theta_{ik} \quad \text{for} \quad j = 1, \ldots, n-1, \text{ and}$$
$$b_{ij} = \prod_{k=1}^{j-1} \sin\theta_{ik} \quad \text{for} \quad j = n.$$

By constructing $\mathbf{B}$ in this manner (with an arbitrary set of angles $\{\theta_{ij}\}$), the resulting $\hat{\mathbf{C}}$ matrix satisfies the conditions for being a valid correlation matrix. Indeed, the trigonometric relationship and the unit radius of the hypersphere, ensure that the ones criterion is adhered to along the main diagonal (see Chapter 3). After choosing a suitable error metric $\epsilon = \|\mathbf{C} - \hat{\mathbf{C}}\|$ (typically the Frobenius norm), an optimisation procedure can be used over the angles θ_{ij}, to find the best possible fit given the chosen error metric (Rebonato and Jäckel, 1999). These procedures are typically algorithms such as Newton-Raphson or conjugate gradient, which are applied to the "angle" space (Pietersz and Groenen, 2004).

Even though angles parametrisation is relatively simple to perform, it does have the proven disadvantages of being less computationally and numerically efficient than other methods proposed in the literature (Pietersz and Groenen, 2004). One of its major advantages, however, is its ability to handle general weights.

2.4.3 Majorisation

The method of majorisation, in the context of low-rank approximations, has emerged as one of the leading techniques in this area of research (Pietersz and Groenen, 2004). It is computationally efficient and provides robust solutions, making it a popular choice amongst practitioners. The idea of majorisation is to find a simpler function than the objective, which has equal function value at a supporting point $\mathbf{y}$. Additionally, this simpler function should be greater than or equal to the objective

function to be minimised, for all points in the domain (Pietersz and Groenen, 2004). This simpler function is termed the *majorisation function* and it is this that we try to minimise, in an iterative manner. This function is typically much simpler than the objective function, hence providing an easier method to obtain the next point in the iteration scheme. The utility of majorisation lies in the fact that it guarantees non-increasing function values along each successive point generated by the algorithm. Furthermore, assuming that both the objective and majorisation functions are, at least, once continuously differentiable (Pietersz and Groenen, 2004), it follows that they have equal gradients at the supporting point $\mathbf{y}$. Hence, the iterative nature of the technique ensures that the next point found will have a strictly smaller objective function value. The diagram below demonstrates the idea of majorisation:

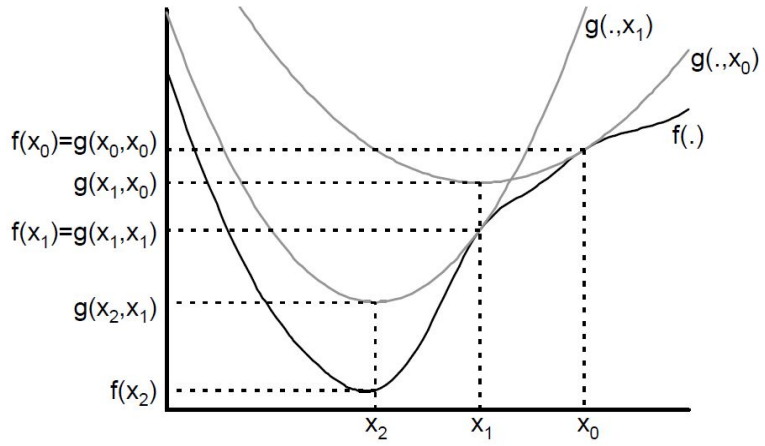


Fig. 2.4: A graphical description of majorisation (*Source*: Pietersz and Groenen (2004))

As can be seen in Figure 2.4, the algorithm begins at $\mathbf{x}_0$, with the majorisation function $g(\cdot, \mathbf{x}_0)$ being fitted by matching the gradient and function value of $f(\cdot)$ at $\mathbf{x}_0$. The function $g(\cdot, \mathbf{x}_0)$ is then subsequently minimised to locate the next point $\mathbf{x}_1$. The process is continued until the minimum is found (which occurs at $\mathbf{x}_2$ in the example above).

Pietersz and Groenen (2004) formalise the procedure as follows: Let $f(\cdot)$ denote the objective function to be minimised. Let there exist a given majorisation function $g(\cdot, \mathbf{y})$ (for each $\mathbf{y}$ in the domain of f) such that,

1. $f(\mathbf{x}) = g(\mathbf{x}, \mathbf{x})$
2. $f(\mathbf{x}) \leq g(\mathbf{x}, \mathbf{x})$ for all x, and
3. $g(\cdot, \mathbf{y})$ is "simple" i.e. the minimum of $g(\cdot, \mathbf{y})$ is simple to compute.

A majorisation algorithm is then given by:

1. Start at $\mathbf{x}^{(0)}$. Set $k := 0$.
2. Set $\mathbf{x}^{(k+1)}$ to the minimum argument of the function $g(\cdot, \mathbf{x}^{(k)})$.

3. If $f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) < \epsilon$ then stop with $\mathbf{x} := \mathbf{x}^{(k+1)}$.

4. Set $k := k + 1$ and repeat from step 2.

It has been shown in Pietersz and Groenen (2004) that the algorithm converges to a stationary point, from any starting point. It is further shown that majorisation is more efficient that any of the previous methods in the literature, including those of the Lagrange multipliers of Zhang and Wu (2003) and the angles parametrisation of Rebonato and Jäckel (1999). Being easier to implement than the other methods (except modified PCA) it has the added advantage of being able to handle arbitrary weights. The full algorithm, with detailed analysis, can be found in Pietersz and Groenen (2004).

2.5 Low-Rank Matrix Completion

Consider the scenario where an individual observes only a small subset of entries from a large matrix, with the remainder of the entries entirely unknown. If this individual aims to recover the full matrix and determine the unknown entries, in the most accurate sense, then this problem falls precisely into the realm of low-rank matrix completion. Low-rank completion is a problem that has been encountered in many areas, from signal processing, recommender systems and machine learning (Chi, 2018). The following sections aim to provide the appropriate justifications and mathematical formalism behind this approach, and draw significantly from the works of Candès and Recht (2009), Candès and Tao (2010) and Chi (2018) in this active area of research.

2.5.1 Eligible Matrix Forms for Completion

In Candès and Recht (2009), it is shown that there are only certain types of matrices for which completion may be successfully performed. If a matrix does not obey certain properties or does not possess a particular structure, then it is impossible to accurately recover the missing elements. Consider the following example, with matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, adapted from Chi (2018):

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{M}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

As most of the entries of $\mathbf{M}_2$ are zero, it is more difficult to complete, as more measurements are required to ensure sufficient mass comes from the non-zero entries (Chi, 2018). In the case of $\mathbf{M}_1$, information propagation is easier from one entry to another, due to the uniformly distributed nature of the matrix, thereby making completion easier.

Another important aspect to note is that there are only certain types of observation patterns that are permitted. Even in a matrix with most of the entries observed, and very few missing elements, there are cases where completion is impossible.

Consider the matrix $\mathbf{M}$ below,

$$\mathbf{M} = \begin{bmatrix} * & ? & * & * \\ * & ? & * & * \\ * & ? & * & * \\ * & ? & * & * \end{bmatrix},$$

where '$*$' indicates an observed entry and '?' indicates an unknown entry (example adapted from Chi (2018)). The second column of unknown entries simply cannot be recovered, as it is free to lie anywhere in the column space of the low-rank matrix. Thus, for a rank-r low-rank matrix, there must be at least r many observations per column/row (Chi, 2018), for valid completion to be possible. In some instances, it is useful to consider random observation patterns that are sampled from some underlying model; the Bernoulli model being a popular example. Here, each entry is identically and independently observed, with probability $p := \frac{m}{n_1 n_2}$. Under this framework, the low-rank recovery of a matrix with less than some $C\mu nr \log n$ observations is impossible, where C is some positive constant, μ is the coherence parameter (defined below), and n is the maximum matrix dimension (Chi, 2018). This quantity is known as the information-theoretic lower bound.

2.5.2 Coherence

The next important notion to consider is that of coherence. The coherence of a matrix is a measure of the alignment of the row/column spaces with the standard basis vectors, in a low-rank matrix (Candès and Recht, 2009). In general, completion of a low-rank matrix is easier, if its energy is evenly distributed across different coordinates, and it is precisely this property that is captured by the coherence. Candès and Recht (2009) and Chi (2018) define coherence as follows: For a matrix $\mathbf{U} \in \mathbb{R}^{n_1 \times r}$ with orthonormal columns, let $\mathbf{P_U}$ be the orthogonal projection onto the column space of $\mathbf{U}$. The coherence parameter of $\mathbf{U}$ is then defined as

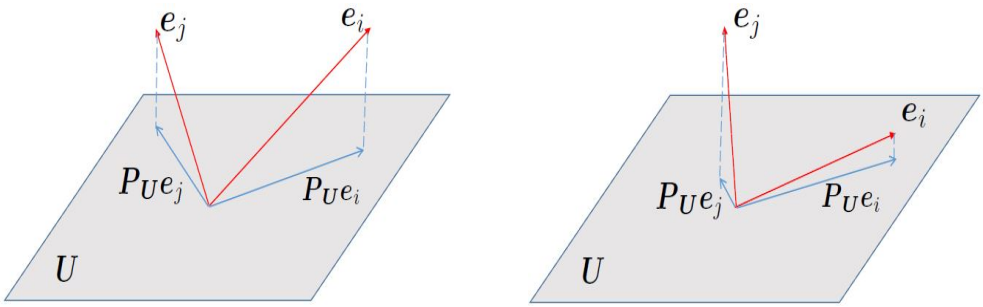
$$\mu(\mathbf{U}) = \frac{n_1}{r} \max_{1 \le i \le n_1} \|\mathbf{P_U e}_i\|_2^2 = \frac{n_1}{r} \max_{1 \le i \le n_1} \|\mathbf{U}^\top \mathbf{e}_i\|_2^2,$$

where $\mathbf{e}_i$ is the ith standard basis vector. For a low-rank matrix $\mathbf{M}$, whose SVD is given by (2.4), the coherence $\mathbf{M}$ is defined as

$$\mu = \max\{\mu(\mathbf{U}), \mu(\mathbf{V})\}.$$

In Figure 2.5, a geometric illustration of the coherence parameter $\mu(\mathbf{U})$ is given.

It can be seen that $\mu(\mathbf{U})$ is small when all the standard basis vectors $\mathbf{e}_i$ have approximately the same projections onto the subspace $\mathbf{U}$, as seen in Figure 2.5a. Conversely, $\mu(\mathbf{U})$ is large if there is too much alignment between $\mathbf{U}$ and certain standard basis vectors, as seen in Figure 2.5b. It should be noted that the singular vectors of $\mathbf{M}$ determine the coherence μ, with μ being independent of the singular values of $\mathbf{M}$ (Chi, 2018). Since $1 \le \mu(\mathbf{U}) \le \frac{n_1}{r}$ and $1 \le \mu(\mathbf{V}) \le \frac{n_2}{r}$, then it follows that $1 \le \mu \le \frac{n}{r}$. In the above example, with matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, we see that the coherence of $\mathbf{M}_1$ matches the lower bound of one, while the coherence of $\mathbf{M}_2$ matches the upper bound of $\frac{n}{r}$ (Chi, 2018). It follows that the smaller the value of μ, the easier it is to perform the completion.



(a) Small coherence. **(b)** Large coherence.

Fig. 2.5: Geometric representation of the coherence parameter $\mu(\mathbf{U})$ (*Source*: Chi (2018)).

2.5.3 Completion by Convex Relaxation

The first technique is completion based on convex relaxation. This is a convex optimisation algorithm, which is both simple and effective in dealing with low-rank completion problems (Jiang and Lyu, 2017). Assume we have a rectangular, $n_1 \times n_2$ matrix $\mathbf{M}$ with partially observed entries, and there exists a target matrix $\mathbf{\Psi}$ that has the missing entries filled in. The first assumption is that $\mathbf{M}$ is of low-rank (the value of which need not be specified) and secondly, that it has the permissible observation patterns and structure as outlined above. We begin by defining a set Ω which contains the observed entries in $\mathbf{M}$. A matrix $P_\Omega(\mathbf{M})$ may then be defined, which takes all the observed entries in $\mathbf{M}$ and sets the missing entries to zero (Jiang and Lyu, 2017). This can be expressed as follows:

$$\Omega = \{(i,j)|(i,j) \in \mathbf{M}\}$$

$$[P_\Omega(\mathbf{M})]_{ij} = \begin{cases} \mathbf{M}_{ij} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

The goal is to recover $\mathbf{M}$ from $P_\Omega(\mathbf{M})$, when the number of observations $m = |\Omega|$ is much smaller than the total number of entries in $\mathbf{M}$. Further, let $n = \max\{n_1, n_2\}$.

Now, as noted in Chi (2018), in order to promote the low-rank structure of the solution, a natural heuristic requires obtaining a matrix of minimum rank, that is still consistent with the given set of observations. The problem can thus be formulated as

$$\min_{\mathbf{\Psi} \in \mathbb{R}^{n_1 \times n_2}} \text{rank}(\mathbf{\Psi}) \quad \text{s.t} \quad P_\Omega(\mathbf{\Psi}) = P_\Omega(\mathbf{M}). \tag{2.8}$$

It turns out, however, that the above formulation is intractable as rank minimisation is NP-hard (Candès and Recht, 2009). Non-deterministic polynomial-time hardness (or NP-hardness) refers to a class of problems that are computationally demanding, and as such, often not practical to solve. In general, no efficient algorithms exist to solve problems such as these, and they often cannot yield exact solutions. This is in contrast to problems that are easily solved in polynomial-time, which are much preferred in terms of computation. In order to bypass performing

a rank minimisation, Candès and Recht (2009) motivate the convex relaxation of the rank heuristic. It is a proven result, that the nuclear norm of a matrix is the tightest convex relaxation of the rank constraint (Fazel, 2002). In other words, the nuclear norm ball $\{\boldsymbol{\Psi} : \|\boldsymbol{\Psi}\|_* \leq 1\}$ is the convex hull of the collection of unit-norm rank-1 matrices: $\{\mathbf{u}\mathbf{v}^\top : \|\mathbf{u}\| = \|\mathbf{v}\| = 1\}$ (Chi, 2018). This result is formalised in the following theorem (see Appendix C for proof), as presented in Fazel (2002):

Theorem 2.12 (Convex Envelope of Rank)**.** *Consider an $m \times n$ matrix $\boldsymbol{X}$. On the set $\mathcal{S} = \{\boldsymbol{X} \in \mathbb{R}^{m\times n} : \|\boldsymbol{X}\| \leq 1\}$, the convex envelope of the function $\phi(\boldsymbol{X}) = \text{rank}(\boldsymbol{X})$ is $\phi_{\text{env}}(\boldsymbol{X}) = \|\boldsymbol{X}\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(\boldsymbol{X})$, where σ_i are the singular-values of $\boldsymbol{X}$.*

It is further noted in both Candès and Recht (2009) and Chi (2018), that due to the nuclear norm being unitarily invariant, it can be represented as the solution to a semi-definite program,

$$\|\boldsymbol{\Psi}\|_* = \min_{\mathbf{Z}_1,\mathbf{Z}_2} \frac{1}{2}(\text{Tr}(\mathbf{Z}_1) + \text{Tr}(\mathbf{Z}_2)) \quad \text{s.t} \quad \begin{bmatrix} \mathbf{Z}_1 & \boldsymbol{\Psi} \\ \boldsymbol{\Psi}^\top & \mathbf{Z}_2 \end{bmatrix} \geq 0.$$

Thus, following the above arguments and invoking Theorem 2.12, we may convert (2.8) from a problem of rank minimisation to one of nuclear norm minimisation. This allows (2.8) to be re-written as:

$$\min_{\boldsymbol{\Psi}\in\mathbb{R}^{n_1\times n_2}} \|\boldsymbol{\Psi}\|_* \quad \text{s.t} \quad P_\Omega(\boldsymbol{\Psi}) = P_\Omega(\mathbf{M}). \tag{2.9}$$

Hence, the above formulation is a convex program, which can now be efficiently solved in polynomial time (Chi, 2018), thereby removing the NP-hardness. In work done by Candès and Tao (2010) and Candès and Plan (2010), amongst others, it is suggested that nuclear norm minimisation can exactly recover a low-rank matrix, as soon as the number of observations exceeds the information-theoretic lower bound by a logarithmic factor (Chi, 2018). Suppose the probability with which each of the entries of $\mathbf{M}$ are observed is $p \in (0, 1)$. Now, if p satisfies

$$p \geq C\frac{\mu r \log^2(n)}{n},$$

for some large enough constant $C > 0$, then (2.9) recovers $\mathbf{M}$ exactly (Chi, 2018), as the unique optimal solution of (2.9) (with high probability). For a sufficiently large number of observations, the geometry of nuclear norm minimisation is demonstrated in Figure 2.6. The level sets of the nuclear norm are represented by the cylinder, whilst the hyperplane represents the measurement constraint. The low-rank solutions that we seek, occur at the intersection of the two sets at the thickened edges. Chi (2018) goes on to note that low-rank matrices can still be recovered, even with a vanishingly small proportion of observations, provided that both μ and r are much smaller than n.

It is an additional fact, proved by Candès and Recht (2009), that "if a matrix has row and column spaces that are incoherent with the standard basis, then nuclear norm minimisation can recover this matrix from a random sampling of a small number of entries."

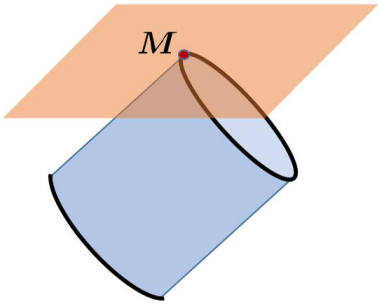


Fig. 2.6: Nuclear norm minimisation represented geometrically (*Source*: Chi (2018)).

2.5.4 Completion by Non-Convex Optimisation

As effective as nuclear norm minimisation is in achieving accurate completion, it suffers from significant computational and memory complexities for large-scale problems (Chi, 2018). This is primarily due to the optimisation over and storing of the matrix variable $\mathbf{\Psi}$ (in the above example). As this process is heavily dependent on n, alternative approaches need to be considered whose complexities are more favourably suited to these higher-dimensional systems. The second approach presented is completion by non-convex optimisation, which is based on gradient descent and utilises a proper initialisation. If we assume that the rank of matrix $\mathbf{M}$ is known, a rank-constrained least-squares problem may be formulated, which incorporates this knowledge of the rank (Chi, 2018):

$$\min_{\mathbf{\Psi}\in\mathbb{R}^{n_1\times n_2}} \|P_\Omega(\mathbf{\Psi}-\mathbf{M})\|_F^2 \quad \text{s.t} \quad \text{rank}(\mathbf{\Psi}) \leq r, \tag{2.10}$$

If we now invoke a low-rank factorisation of the form $\mathbf{\Psi} = \mathbf{X}\mathbf{Y}^\top$, where $\mathbf{X} \in \mathbb{R}^{n_1\times r}$ and $\mathbf{Y} \in \mathbb{R}^{n_2\times r}$, (2.10) can be rewritten as the following unconstrained, non-convex optimisation problem (Chi, 2018):

$$\min_{\mathbf{X},\mathbf{Y}} f(\mathbf{X},\mathbf{Y}) := \|P_\Omega(\mathbf{X}\mathbf{Y}^\top - \mathbf{M})\|_F^2, \tag{2.11}$$

Chi (2018) notes that even though the memory complexities of $\mathbf{X}$ and $\mathbf{Y}$ are now linear in n (as opposed to quadratic when considering $\mathbf{\Psi}$), they can only be determined up to the scaling and rotational ambiguities in (2.11). For an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{r\times r}$ and for any $\alpha \neq 0$, we have $\mathbf{X}\mathbf{Y}^\top = (\alpha\mathbf{X}\mathbf{Q})(\alpha^{-1}\mathbf{Y}\mathbf{Q})^\top$. In order to correct this scaling ambiguity, Chi (2018) proposes the following modified loss function:

$$F(\mathbf{X},\mathbf{Y}) = \frac{1}{4p} f(\mathbf{X},\mathbf{Y}) + \frac{1}{16}\|\mathbf{X}^\top\mathbf{X} - \mathbf{Y}^\top\mathbf{Y}\|_F^2, \tag{2.12}$$

where p is the observation probability and the second term is included to account for solutions where $\mathbf{X}$ and $\mathbf{Y}$ have balanced norms. It should be noted that in cases where p in unknown, a sufficiently good estimate is the sample proportion $\frac{|\Omega|}{n_1 n_2}$ (Chi, 2018). Further details on this non-convex loss function $F(\mathbf{X},\mathbf{Y})$, together with a more detailed optimisation procedure can be found in Appendix B.

Chapter 3

Constructing Real Correlation Matrices

The first aspect to consider, before addressing the problem of matrix completion, is the construction of real correlation matrices for the purposes of testing. There exists a number of full-rank parametrisations in the literature, each with their own particular structure and utility. In general, there are three requirements for a matrix to be considered a valid correlation matrix:

1. The matrix must be symmetric with valid correlations as entries i.e. $-1 \leq \rho \leq 1$.

2. The main diagonal needs to be a vector of ones i.e. the ones criterion (Zhang and Wu, 2003).

3. The matrix needs to be positive semi-definite (non-negative eigenvalues).

For notational simplicity, matrices with the above properties of being symmetric, positive semi-definite will be referred to as being "sPSD". It should be noted that if it is known that a correlation matrix is sPSD, and if some eigenvalues are found to be zero, then it can be immediately determined that the system is not full-factored, i.e. it is of low-rank. In the construction of correlation matrices, the following definition is also of importance.

Definition 3.1 (Number of Correlations)**.** The number of correlations, N_0, in an $n \times n$ correlation matrix is given by

$$N_0 = \frac{n(n-1)}{2},$$

where n is the dimension of the matrix.

Brigo and Mercurio (2006) provide a handful of useful full-rank parametrisations, with two and three-parameter exponential forms given. For the purposes of testing, the classical, two-parameter, exponentially decreasing parametrisation was chosen (Brigo and Mercurio, 2006):

$$\rho_{i,j} = \rho_\infty + (1 - \rho_\infty) \exp(-\beta |i - j|), \tag{3.1}$$

where ρ_∞ represents the correlation between the farthest elements in the system.

There are, however, some limitations in the above parametrisation, primarily being its inability to incorporate noise. This concept of noise is very important, as

it is a key feature of real-world data and is a defining characteristic of a real correlation matrix. Hardin *et al.* (2013) provide an overview of three existing models, with a novel technique of incorporating noise into these matrices in a highly controlled manner. These three models are namely: the constant correlation model, the hub-observation model and the Toeplitz model. The latter was chosen for the purposes of testing, as it is a popular classification which assigns higher correlations to adjacent pairs of observations and lower correlations to those further away, and is widely-used in time-series data analysis. Hardin *et al.* (2013) propose a matrix of the form:

$$\Sigma_k = \begin{pmatrix} 1 & \rho_k & \rho_k^2 & \rho_k^3 & \cdots & \rho_k^{g_k-1} \\ \rho_k & 1 & \rho_k & \rho_k^2 & \cdots & \rho_k^{g_k-2} \\ \rho_k^2 & \rho_k & 1 & \rho_k & \cdots & \rho_k^{g_k-3} \\ \rho_k^3 & \rho_k^2 & \rho_k & 1 & \cdots & \rho_k^{g_k-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_k^{g_k-1} & \rho_k^{g_k-2} & \rho_k^{g_k-3} & \rho_k^{g_k-4} & \cdots & 1 \end{pmatrix}$$

Fig. 3.1: Toeplitz structure as proposed in Hardin *et al.* (2013).

It should be noted that the two parametrisations discussed above, were chosen as the main test-cases, on which the matrix completion algorithms (discussed in Chapter 4) were tested.

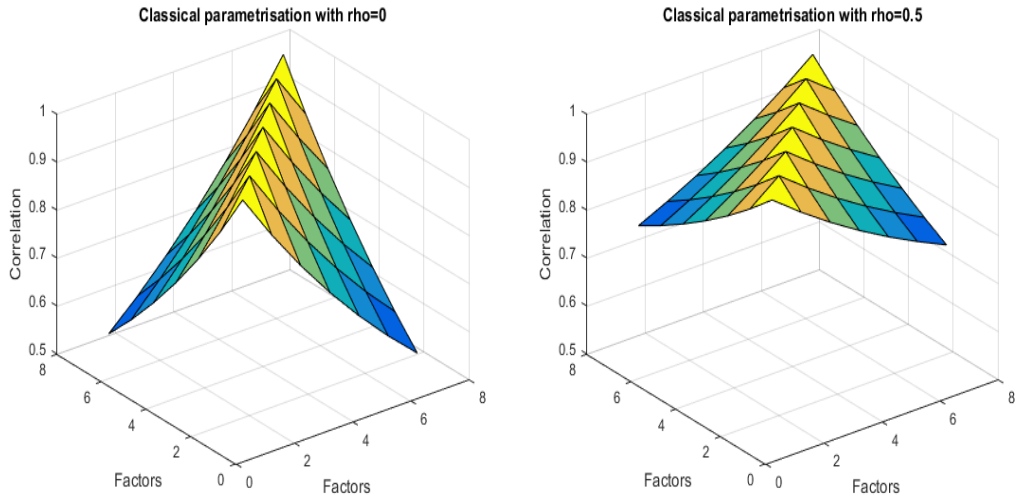
Using the algorithm proposed in Hardin *et al.* (2013), with a matrix group size of $g_k = 7$ and noise-space dimension $M = 3$, the following 7×7 correlation matrix was constructed:

$$C = \begin{pmatrix} 1.00000 & 0.72722 & 0.57393 & 0.46403 & 0.38641 & 0.35441 & 0.22740 \\ 0.72722 & 1.00000 & 0.89601 & 0.71983 & 0.59831 & 0.41658 & 0.41590 \\ 0.57393 & 0.89601 & 1.00000 & 0.89445 & 0.72861 & 0.56205 & 0.51850 \\ 0.46403 & 0.71983 & 0.89445 & 1.00000 & 0.84425 & 0.71408 & 0.59847 \\ 0.38641 & 0.59831 & 0.72861 & 0.84425 & 1.00000 & 0.82224 & 0.74916 \\ 0.35441 & 0.41658 & 0.56205 & 0.71408 & 0.82224 & 1.00000 & 0.85190 \\ 0.22740 & 0.41590 & 0.51850 & 0.59847 & 0.74916 & 0.85190 & 1.00000 \end{pmatrix}.$$

The next matrix was constructed using (3.1), which is the classical parametrisation of Brigo and Mercurio (2006), with $\rho_\infty = 0$:

$$C = \begin{pmatrix} 1.00000 & 0.90484 & 0.81873 & 0.74082 & 0.67032 & 0.60653 & 0.54881 \\ 0.90484 & 1.00000 & 0.90484 & 0.81873 & 0.74082 & 0.67032 & 0.60653 \\ 0.81873 & 0.90484 & 1.00000 & 0.90484 & 0.81873 & 0.74082 & 0.67032 \\ 0.74082 & 0.81873 & 0.90484 & 1.00000 & 0.90484 & 0.81873 & 0.74082 \\ 0.67032 & 0.74082 & 0.81873 & 0.90484 & 1.00000 & 0.90484 & 0.81873 \\ 0.60653 & 0.67032 & 0.74082 & 0.81873 & 0.90484 & 1.00000 & 0.90484 \\ 0.54881 & 0.60653 & 0.67032 & 0.74082 & 0.81873 & 0.90484 & 1.00000 \end{pmatrix}.$$

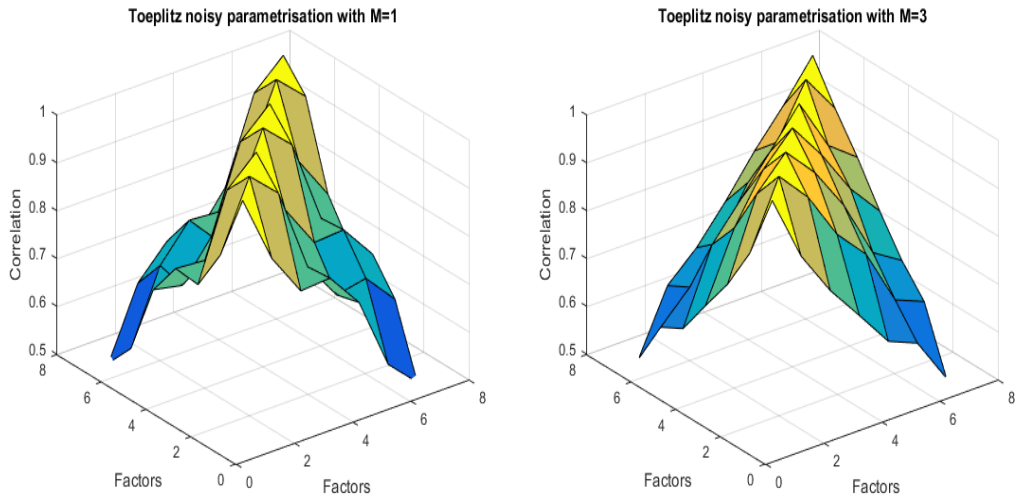
Figures 3.2 and 3.3 are surface plots of the effect of different ρ_∞ and noise-space dimension values (M) respectively.



(a) Using $\rho_\infty = 0$.

(b) Using $\rho_\infty = 0.5$.

Fig. 3.2: Classical parametrisation with different ρ_∞ values.



(a) Using noise-space dimension $M = 1$.

(b) Using noise-space dimension $M = 3$.

Fig. 3.3: Noisy Toeplitz parametrisation with different noise-space values.

Chapter 4

The Completion Problem

To better understand the completion problem, a suitable starting point would be to first frame the problem at hand:

A user provides an $N \times N$ correlation matrix of a given low-rank n. This user-defined rank has been determined to be sufficient for the modelling purposes of the user. Suppose that d additional factors are introduced into the system, resulting in a new $(N+d)\times(N+d)$ correlation matrix, with $\frac{d}{2}(d+2N-1)$ additional correlation terms. How well can the $(N+d) \times (N+d)$ matrix be completed (i.e. the missing correlations estimated), to yield the most accurate rank-n correlation matrix and is this consistent with preserving the original user-defined $N \times N$ matrix of rank-n?

In essence, this problem may be considered from two aspects. The first is to explore whether the completion can be achieved whilst maintaining the original user-defined $N \times N$ matrix and how closely this approximates the theoretical, optimal low-rank rank-n $(N+d) \times (N+d)$ matrix. The second aspect to then consider, is whether an improved approximation can be achieved (which is closer to the theoretical, optimal low-rank matrix) by allowing the original user-defined $N \times N$ matrix to change.

4.1 Preserving the Original User-Defined Matrix

Referring to the left side of Figure 4.1, this rank-reduced $N \times N$ sub-matrix is augmented with the additional d factors, which includes sparsely observed correlations, ρ_i, within the $(N+d) \times (N+d)$ matrix. We now present the first completion method, which we have termed *spectral completion*. This is an indirect method, which automatically accounts for the symmetric nature of the correlation structure and relies on the use of the square-root matrix. Assuming that this full matrix, $\mathbf{W}$, (on the left side of Figure 4.1) is a valid correlation matrix (sPSD), it may be decomposed (via the spectral theorem) into the following form $\mathbf{W} = \bar{\mathbf{B}}\bar{\mathbf{B}}^\top$. Here, $\bar{\mathbf{B}}$ is an $(N+d) \times n$ matrix, where n is the required rank. The matrix $\bar{\mathbf{B}}$ is composed of two sub-matrices, $\mathbf{B}$ and $\mathbf{b}$. The elements of $\mathbf{B}$ are all known, as they form part of the decomposition of the given $N \times N$ matrix. The $\mathbf{b}$ matrix, however, contains $n \times d$ unknowns, which need to be approximated, thereby completing matrix $\bar{\mathbf{B}}$. Hence, multiplying this matrix by its transpose, will yield a completed, approximated $(N+d)\times(N+d)$ matrix $\mathbf{W}$ of rank-n. Now, in order to calculate the missing entries of $\mathbf{b}$, a system of linear equations may be set up, which relates the rows of

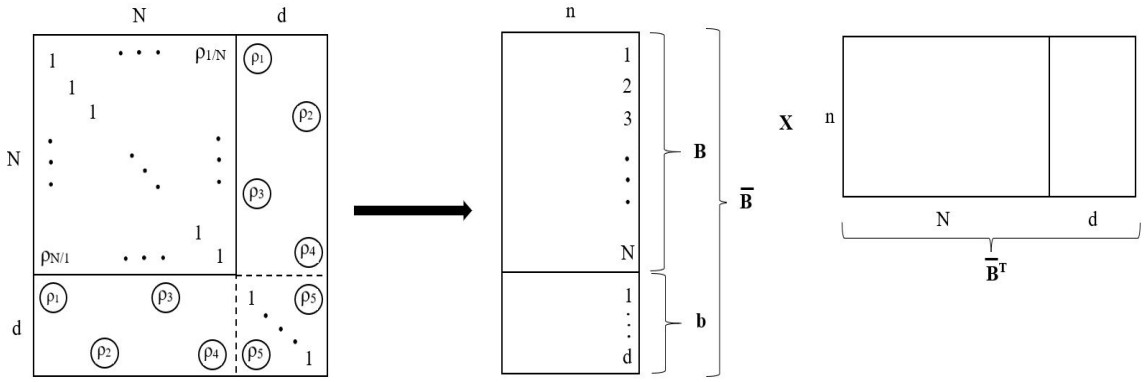


Fig. 4.1: Decomposition in the square-root matrix

B and **b** (equivalently the full $\bar{\mathbf{B}}$ matrix), to input correlations in the initial, incomplete **W** matrix. Note that it is assumed that only a sparse set of input correlations have been given, with the aim of completing the missing entries. For a given, off-diagonal correlation $\rho_{x/y}$ and main diagonal correlation $\rho_{x/x} = 1$, equations of the following form may be constructed and solved:

$$\bar{\mathbf{B}}_x \cdot \bar{\mathbf{B}}_y^\top = \rho_{x/y}$$
$$\bar{\mathbf{B}}_x \cdot \bar{\mathbf{B}}_x^\top = 1$$

Consider the following example: A rank-3 5×5 correlation matrix is provided by the user, which has been previously determined by some means. Now, three additional factors are added to the system, with only certain cross-correlations having been determined. In this new 8×8 correlation matrix, assume that only $\rho_{1/6}$, $\rho_{3/6}$, $\rho_{5/8}$, $\rho_{4/7}$, $\rho_{6/7}$ and $\rho_{6/8}$ are known. This leads to the following system of equations which must be solved (refer to Figure 4.1),

$$\bar{\mathbf{B}}_1 \cdot \bar{\mathbf{B}}_6^\top = \mathbf{B}_1 \cdot \mathbf{b}_1^\top = \rho_{1/6}$$
$$\bar{\mathbf{B}}_3 \cdot \bar{\mathbf{B}}_6^\top = \mathbf{B}_3 \cdot \mathbf{b}_1^\top = \rho_{3/6}$$
$$\bar{\mathbf{B}}_5 \cdot \bar{\mathbf{B}}_8^\top = \mathbf{B}_5 \cdot \mathbf{b}_3^\top = \rho_{5/8}$$
$$\bar{\mathbf{B}}_4 \cdot \bar{\mathbf{B}}_7^\top = \mathbf{B}_4 \cdot \mathbf{b}_2^\top = \rho_{4/7}$$
$$\bar{\mathbf{B}}_6 \cdot \bar{\mathbf{B}}_7^\top = \mathbf{b}_1 \cdot \mathbf{b}_2^\top = \rho_{6/7}$$
$$\bar{\mathbf{B}}_6 \cdot \bar{\mathbf{B}}_8^\top = \mathbf{b}_1 \cdot \mathbf{b}_3^\top = \rho_{6/8},$$

subject to the constraints,

$$\bar{\mathbf{B}}_6 \cdot \bar{\mathbf{B}}_6^\top = \mathbf{b}_1 \cdot \mathbf{b}_1^\top = 1 \tag{4.1}$$
$$\bar{\mathbf{B}}_7 \cdot \bar{\mathbf{B}}_7^\top = \mathbf{b}_2 \cdot \mathbf{b}_2^\top = 1 \tag{4.2}$$
$$\bar{\mathbf{B}}_8 \cdot \bar{\mathbf{B}}_8^\top = \mathbf{b}_3 \cdot \mathbf{b}_3^\top = 1. \tag{4.3}$$

By solving this system of equations, the unknown matrix **b** and subsequently the matrix $\bar{\mathbf{B}}$ will be completed, and multiplying by its transpose will hence yield a

rank-3 approximated 8×8 matrix.

The general algorithm for testing is outlined below:

1. A theoretical $(N+d) \times (N+d)$ correlation matrix, $\mathbf{R}$ was constructed (classical formulation or the noisy Toeplitz structure) and an eigenvalue decomposition of the form $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ (with quantities as per (2.2)) was performed. The best rank-reduced rank-n correlation matrix, $\mathbf{S}$, was then found by applying modified PCA (or some other rank-reduction technique).

2. An $H \times (N+d)$ matrix $\mathbf{U}$, of standard normal random variates was generated, where H is the number of samples. Using the square-root matrix $\mathbf{V} = \mathbf{Q}\mathbf{\Lambda}^{1/2}$, a correlated matrix of random variates $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$ was generated.

3. From this data, the full-rank sample correlation matrix, $\mathbf{R'}$, was generated by finding the cross-correlations (normal PCA can be applied as per Appendix A), after which modified PCA was applied to find the best rank-reduced rank n approximated correlation matrix, $\mathbf{S'}$, of size $(N+d) \times (N+d)$.

4. The $N \times N$ sub-matrix of $\mathbf{R'}$ was extracted and modified PCA was again applied to rank-reduce it to rank-n.

5. A few cross-correlations between the original N factors and additional d factors from the full-rank $\mathbf{S'}$ matrix (including one or more intra-correlations between the d factors themselves) were chosen, resulting in an incomplete matrix, similar to the one on the left side of Figure 4.1.

6. The matrix completion was performed by formulating the problem in terms of a square-root decomposition, and an optimisation routine was used (*fmincon* in MATLAB) to solve the system of equations.

7. This low-rank, approximated matrix, $\mathbf{C}$, was then compared to the theoretical, optimal low-rank matrix, $\mathbf{S}$, and the resulting MNE and ENE metrics were compared (as defined in Section 2.1).

Note, the above algorithm describes, in general, the method for approaching the problem directly in terms of correlations. It may be approached indirectly by first working with covariances, and then later recovering the correlation matrices, with the general algorithm being very similar (discussed later).

4.1.1 Correlation Method (Constrained Optimisation)

As mentioned above, the first approach formulated the problem directly in terms of correlations. This, however, required a constrained optimisation, as the unknown entries of matrix $\mathbf{b}$ must be solved such that $\mathbf{b}_x\mathbf{b}_x^\top = 1$ (where $\mathbf{b}_x$ is the xth row of $\mathbf{b}$), so that the ones criterion is preserved along the main diagonal. Furthermore, the resulting matrix must remain positive, semi-definite, which is critical for maintaining a valid correlation structure. The constrained optimiser, *fmincon*, of MATLAB was used as the primary solver for the optimisation problem. This function utilises

a gradient descent algorithm, as part of the iterative optimisation process. It was then used to minimise an objective function that was set up, such that the given constraints were satisfied.

With reference to the earlier example, this objective function was formulated by rearranging the correlation relationships between the given correlation estimates and the rows of matrix $\bar{\mathbf{B}}$. Thus, a vector of the following form was created:

$$X = \left(\begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{B}}_3 \\ \bar{\mathbf{B}}_5 \\ \bar{\mathbf{B}}_4 \\ \bar{\mathbf{B}}_6 \\ \bar{\mathbf{B}}_6 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{B}}_6 \\ \bar{\mathbf{B}}_6 \\ \bar{\mathbf{B}}_8 \\ \bar{\mathbf{B}}_7 \\ \bar{\mathbf{B}}_7 \\ \bar{\mathbf{B}}_8 \end{bmatrix}^\top - \begin{bmatrix} \rho_{1/6} \\ \rho_{3/6} \\ \rho_{5/8} \\ \rho_{4/7} \\ \rho_{6/7} \\ \rho_{6/8} \end{bmatrix} \right).$$

A minimisation in the Euclidean-norm was performed, which minimised the sum of squared differences in the objective function. This is typically more robust and stable, and computationally more efficient that L1-norm minimisation (Bektaş and Şişman, 2010). Using X as above, the final form of the objective function was given as

$$f(X) = X^\top X + W_1 \max(0, -M_{ev}), \tag{4.4}$$

where M_{ev} is the minimum eigenvalue of $\mathbf{C}$ and W_1 is a large, positive weight. The presence of the second term enforces positive semi-definiteness, by ensuring that the eigenvalues of the system remain non-negative. It should be noted that the above optimisation was performed, subject to the direct constraint equations (4.1), (4.2), and (4.3), so that the ones criterion was preserved along the main diagonal, thereby making this method one of constrained optimisation.

4.1.2 Covariance Method (Unconstrained Optimisation)

The second approach taken, involved specifying the completion problem in terms of covariance matrices. By working with covariance structures, the need for constrained optimisation was removed, as the ones criterion no longer needed to be directly satisfied. The unknowns in the decomposed $\bar{\mathbf{B}}$ matrix were now square-root covariance terms, and the resultant matrix consisted of variances along the main diagonal and covariances along the off-diagonals. By allowing the optimiser to perform an unconstrained optimisation (with the objective function formulated as before except now in terms of covariances), a larger solution set was available from which the optimiser could better fit the solution.

$$X = \left(\begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{B}}_3 \\ \bar{\mathbf{B}}_5 \\ \bar{\mathbf{B}}_4 \\ \bar{\mathbf{B}}_6 \\ \bar{\mathbf{B}}_6 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{B}}_6 \\ \bar{\mathbf{B}}_6 \\ \bar{\mathbf{B}}_8 \\ \bar{\mathbf{B}}_7 \\ \bar{\mathbf{B}}_7 \\ \bar{\mathbf{B}}_8 \end{bmatrix}^\top - \begin{bmatrix} \sigma_{1/6} \\ \sigma_{3/6} \\ \sigma_{5/8} \\ \sigma_{4/7} \\ \sigma_{6/7} \\ \sigma_{6/8} \end{bmatrix} \right).$$

The vector X is now specified as above, where $\sigma_{x/y}$ is the covariance between factors x and y. The final objective function was formulated as per (4.4), with the additional term again included to account for positive, semi-definiteness. The approximated low-rank correlation matrix was then simply recovered from the covariance matrix, after extracting the relevant variances and standard deviations.

4.1.3 Impact of Input Correlations to the Algorithm

Another aspect to consider, is the choice of input correlations provided to the algorithm. These had an effect on the matrix completion, as there exists an influence of the underlying correlation structure. Notwithstanding the theoretical considerations given in Section 2.5.1, and in order to achieve reasonable solutions, there needs to be at least one or more intra-correlations provided (i.e. correlations between the additional d factors), as well as inter-correlations between the d and original N factors. In particular, it was found that improved fitting was achieved when the correlations at the upper and left boundaries of the matrix (i.e. row one and column one with the additional d factors) were given, as opposed to being left unspecified.

4.2 Modifying the Original User-Defined Matrix

The approach of spectral completion, while simple and relatively effective, still represents a non-optimal attempt at achieving accurate completion. As the problem at hand involves the low-rank completion of correlation matrices, it was natural to explore the effectiveness of completion by convex relaxation, as outlined in Section 2.5. As noted, this theory was developed for arbitrary, rectangular matrices of low-rank, and not specifically for correlation structures, which were the focus of this dissertation.

The setup for this technique is very similar to that given in Section 4.1, with the same notation being followed. Recall the problem statement that is being addressed:

A user provides an $N \times N$ correlation matrix of a given low-rank n. This user-defined rank has been determined to be sufficient for the modelling purposes of the user. Suppose that d additional factors are introduced into the system, resulting in a new $(N+d)\times(N+d)$ correlation matrix, with $\frac{d}{2}(d+2N-1)$ additional correlation terms. How well can the $(N + d) \times (N + d)$ matrix be completed (i.e. the missing correlations estimated), to yield the most accurate rank-n correlation matrix and is this consistent with preserving the original $N \times N$ matrix of rank-n?

Consider a partially observed, low-rank correlation structure with missing entries:

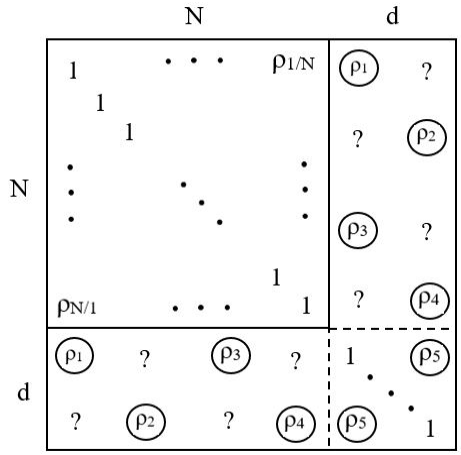


Fig. 4.2: Partially observed low-rank correlation matrix of rank-n.

With reference to Section 2.5.3 and following the same notation, we now formulate the problem specifically for the case of valid correlation matrices (as opposed to the general case).

Assume we have a square, $(N+d) \times (N+d)$ matrix $\mathbf{M}$ with a main diagonal of ones and partially observed entries on the off-diagonals. Further, assume that there exists a target matrix $\mathbf{\Psi}$ that has the missing correlations filled in with guesses. It is known that $\mathbf{M}$ is of some low-rank n and assume it has the permissible observation patterns and structure as outlined in Section 2.5.3. We begin by defining a set Ω which contains the observed entries in $\mathbf{M}$ (including the main diagonal of ones). A matrix $P_\Omega(\mathbf{M})$ may then be defined, which takes all the observed entries in $\mathbf{M}$ and sets the missing entries to zero. Thus, we have the following:

$$\Omega = \{(i,j) | (i,j) \in \mathbf{M}\}$$

$$[P_\Omega(\mathbf{M})]_{ij} = \begin{cases} \mathbf{M}_{ij} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

Recall that the problem is initially specified as one of rank-minimisation. In particular, we require a solution for

$$\min_{\mathbf{\Psi} \in \mathbb{R}^{(N+d)\times(N+d)}} \text{rank}(\mathbf{\Psi}) \quad \text{s.t.} \quad P_\Omega(\mathbf{\Psi}) = P_\Omega(\mathbf{M}). \tag{4.5}$$

With the appropriate justification given in Section 2.5.3 and invoking Theorem 2.12, the problem may be re-formulated in terms of a nuclear norm minimisation as

$$\min_{\mathbf{\Psi} \in \mathbb{R}^{(N+d)\times(N+d)}} \|\mathbf{\Psi}\|_* \quad \text{s.t.} \quad P_\Omega(\mathbf{\Psi}) = P_\Omega(\mathbf{M}). \tag{4.6}$$

In order to perform the minimisation, *fmincon* of MATLAB was utilised. The objective functions corresponding to (4.5) and (4.6) may be expressed simply as (Jiang

and Lyu, 2017),

$$\text{minimise} \;\; \|P_\Omega(\mathbf{\Psi}) - P_\Omega(\mathbf{M})\|_F + k\,\text{rank}(\mathbf{\Psi}),$$

which further simplifies to,

$$\text{minimise} \;\; \|P_\Omega(\mathbf{\Psi}) - P_\Omega(\mathbf{M})\|_F + k\|\mathbf{\Psi}\|_*,$$

for some positive weight k. This is where the formulation of the general low-rank completion theory stops, which suffices for arbitrary low-rank matrices. However, there are certain additional considerations for the special case of correlation structures, namely maintaining positive semi-definiteness, ensuring symmetry and maintaining the ones criterion along the main diagonal. As such, the following modified objective function is proposed:

$$\text{minimise} \;\; \|P_\Omega(\mathbf{\Psi}) - P_\Omega(\mathbf{M})\|_F + k\|\mathbf{\Psi}\|_* + W_1\text{max}(0, -M_{ev}), \tag{4.7}$$

where M_{ev} is the minimum eigenvalue of $P_\Omega(\mathbf{\Psi})$, and W_1 is a large, positive weight. The third term accounts for positive semi-definiteness, by ensuring that the eigenvalues of the system remain non-negative. It should be further noted that $P_\Omega(\mathbf{M})$, and subsequently $P_\Omega(\mathbf{\Psi})$, had the ones criterion enforced by construction, as per the requirements for a valid correlation matrix. Noting this point and by including the additional term in (4.7), the solution space of the optimiser is therefore guaranteed to yield completion results, which result in a valid $(N+d)\times(N+d)$ correlation matrix.

The general algorithm for completion by convex relaxation is given below:

1. A theoretical $(N+d)\times(N+d)$ correlation matrix, **R** was constructed (classical formulation or the noisy Toeplitz structure) and an eigenvalue decomposition of the form $\mathbf{R} = \mathbf{Q\Lambda Q}^\top$ (with quantities as per (2.2)) was performed. The best rank-reduced rank-n correlation matrix, **S**, was then found by applying modified PCA (or some other rank-reduction technique).

2. An $H\times(N+d)$ matrix **U**, of standard normal random variates was generated, where H is number of samples. Using the square-root matrix $\mathbf{V} = \mathbf{Q\Lambda}^{1/2}$, a correlated matrix of random variates $\mathbf{X} = \mathbf{UV}^\top$ was generated.

3. From this data, the full-rank sample correlation matrix, **R'**, was generated by finding the cross-correlations (using normal PCA as per Appendix A), after which modified PCA (or some other rank-reduction technique) was applied to find the best rank-reduced rank n approximated correlation matrix, **S'**, of size $(N+d)\times(N+d)$.

4. The $N\times N$ sub-matrix of **R'** was extracted and modified PCA was again applied to rank-reduce it to rank n.

5. A few cross-correlations between the original N factors and additional d factors from the low-rank **S'** matrix (including one or more intra-correlations

between the d factors themselves) were chosen, resulting in an incomplete matrix, similar to the one in Figure 4.2. This partially observed matrix may be treated as matrix **M**, with missing entries which we require to complete.

6. Nuclear norm minimisation was then performed with the use of *fmincon* and (4.7), thereby completing the matrix by convex relaxation.

7. It should be noted that by applying this technique, the resulting $(N + d) \times (N + d)$ correlation matrix, **C**, is no longer of low-rank n, even though the original $N \times N$ user-defined matrix of rank-n is maintained. Thus, modified PCA was applied to the overall matrix to rank-reduce it to rank-n, thereby resulting in the best low-rank approximated correlation matrix, $\hat{\mathbf{C}}$, from the data. This matrix was then compared to the theoretical, optimal low-rank matrix, **S**, and the resulting MNE and ENE metrics were compared (as defined in Section 2.1).

It is this final step, by performing modified PCA on the completed matrix to rank-reduce it to rank-n, which results in the slight modification of the original $N \times N$ matrix. The following chapter provides testing results, indicating which of the above two techniques, namely spectral completion and completion by convex relaxation, is superior, in terms of most closely approximating the theoretical, optimal low-rank matrix, **S**. Indeed, the results obtained from numerical testing make for interesting analysis.

Chapter 5

Results

To ensure effectiveness of the algorithms previously presented, extensive numerical testing was performed with various combinations of parametric correlation matrices, completion methods and correlation inputs. The first set of testing involved the use of synthetic data and theoretical correlation matrices, to test and compare the effectiveness of each of the completion methods, as expounded in Chapter 4. These were namely the correlation and covariance methods (which collectively fall under spectral completion) and completion by convex relaxation. Once synthetic testing was completed, the completion techniques were then applied to two sets of real market data. The error metrics used in the analysis were the MNE and ENE, as defined in Section 2.1.3. The MNE is an indicator of the overall "closeness" of the approximation, while the ENE is important, as it is a direct indicator of how closely the PCs of the approximated and true matrices are matched. Note that a list of the correlation pairs that were used as inputs to the algorithms (for each of the test-cases outlined below) is given in Appendix D.2. Furthermore, wherever the number of input correlations are mentioned, note that these, in fact, refer to input correlation pairs, due to the symmetry of the correlation structures. For example, if 6 input correlations were said to be provided to the algorithms, this implies that a total of 12 correlations were used as inputs (due to symmetry).

5.1 Synthetic Data Testing

5.1.1 Classical Exponential Form

The first dataset was based upon the classical exponential form, as per (3.1). Two scenarios were considered: the first is a fully-determined case, where the number of input correlations matched the number of unknowns for the spectral completion techniques (see Section 4.1), as the rank increased. The same correlation inputs were then used when applying convex relaxation for consistency of results. The second scenario consisted of an under-determined set of inputs, where fewer inputs were provided than the number of unknowns in the system (for the spectral methods). The same under-determined set of correlations was then provided to the convex relaxation algorithm. In many ways, this testing approach (by using an under-determined system) is more significant, as it tests a worst-case scenario to check the efficacy and robustness of the approximations. It should be noted that all comparisons were made relative to the theoretical, optimal low-rank matrices

(for each rank), which were determined by applying modified PCA to the original, full-rank matrices. These were the optimal low-rank matrices, and our goal was to test how favourably the approximations of the different completion methods, compared to the theoretical "true" optimal matrices. A matrix with the following parameters was constructed: $\rho_\infty = 0$, $N = 5$, $d = 2$ i.e. a 5×5 matrix, with 2 additional factors added to the system, with a varying rank for testing. The first set of results are presented in Tables 5.1 and 5.2.

Tab. 5.1: Fully-determined classical system comparison of completion methods.

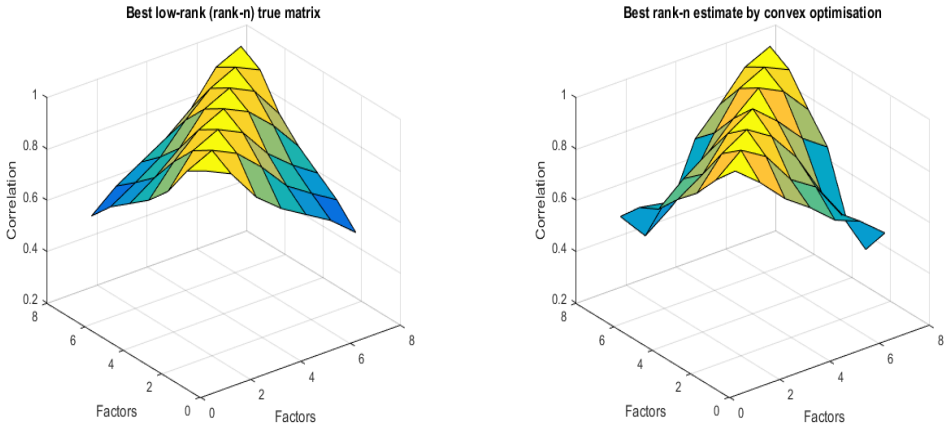
Classical	Correlation Method				Covariance Method				Convex Relaxation			
Rank (n)	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.
2	0.31154	5.012%	0.03193	0.514%	0.39425	6.343%	0.00340	0.055%	0.21196	3.410%	0.04569	0.735%
3	0.64845	10.823%	0.12823	2.140%	0.46221	7.714%	0.08802	1.469%	0.11418	1.906%	0.03717	0.620%
4	0.33402	5.677%	0.08693	1.477%	1.34667	22.889%	8.09251	137.547%	0.07176	1.220%	0.01603	0.272%

As can be seen, the MNEs for both the correlation and covariance methods are similar, however the covariance method did not perform well in the rank-4 case, with an error of 22.9% from the true matrix. In terms of the ENE, both results are extremely good for rank-2 and rank-3 (under 2.5%), however, the covariance method did not perform well once again in the case of rank-4, while the correlation method yields an error of 1.48%. Convex relaxation performed better than the spectral methods, in terms of both the MNE and ENE. In fact, it maintains MNE errors of $< 3.5\%$ across all ranks and ENE errors of $< 1\%$.

Tab. 5.2: Under-determined classical system comparison of completion methods.

Classical	Correlation Method				Covariance Method				Convex Relaxation			
Rank (n)	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.
3	0.61564	10.275%	0.16274	2.716%	0.69549	11.608%	0.35425	5.912%	0.32370	5.403%	0.22203	3.706%
4	0.49145	8.353%	0.20744	3.526%	0.41770	7.100%	0.13075	2.222%	0.31430	5.342%	0.19908	3.384%
5	0.31906	5.487%	0.04540	0.781%	0.37124	6.385%	8.19572	140.953%	0.28639	4.926%	0.14723	2.532%

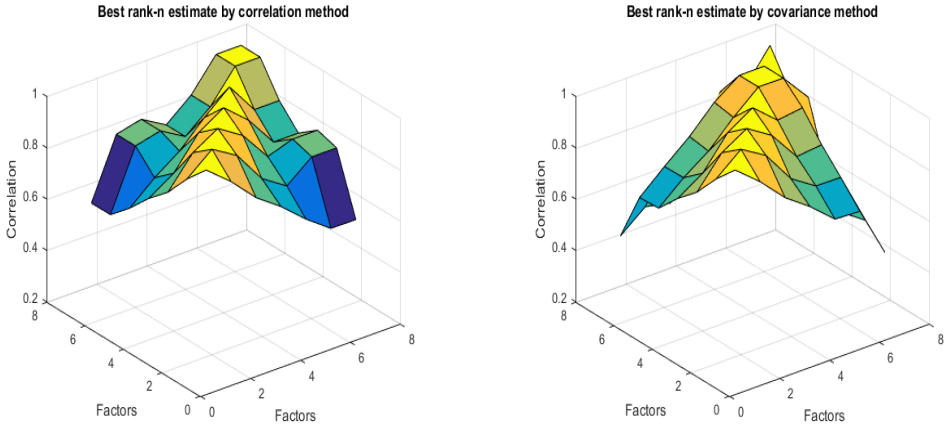
In the under-determined case, we once again see the method of convex relaxation performing better than both of the spectral methods. The MNEs for both the correlation and covariance methods are comparable (across the ranks), while the ENE for the correlation method is far superior to that of the covariance method (all $< 4.0\%$). Convex relaxation yields MNEs of $< 5.5\%$ and ENEs of $< 4.0\%$, which is consistent across all ranks. Note once again that this represents a worst-case type testing, with convex relaxation clearly performing better and more consistently, across different ranks. Figures 5.3 and 5.2, provide a visual representation of the completion methods up to rank-4. As can be seen, convex relaxation fits the true, low-rank matrix more closely than the spectral methods.



(a) Using $\rho_\infty = 0$, $n = 4$.

(b) Convex relaxation method, $n = 4$.

Fig. 5.1: Comparison of true low-rank matrix and convex relaxation approximation for an under-determined classical exponential system



(a) Correlation method, $n = 4$.

(b) Covariance method, $n = 4$.

Fig. 5.2: Comparison of correlation and covariance approximations for an under-determined classical exponential system.

5.1.2 Noisy Toeplitz Form

The next set of tests involved using the noisy Toeplitz parametrisation as the theoretical correlation structure. This is a useful test set, as it allows the controlled inclusion of noise into the matrix, which provides another interesting aspect to the analysis. The procedure was the same as per the classical case, where both fully and under-determined systems were considered, with recovery performed to different ranks. A matrix with the following parameters was generated: $Noise\text{-}space\ (M) = 3$, $N = 5$, $d = 2$.

Tab. 5.3: Fully-determined noisy Toeplitz system comparison of completion methods.

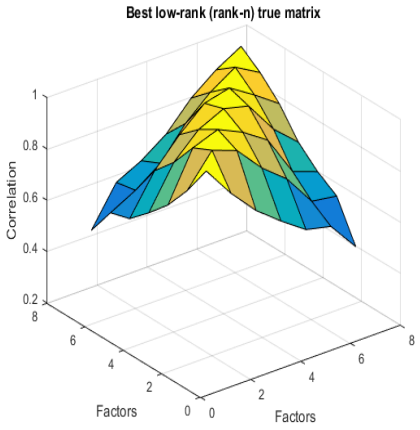
Toeplitz	Correlation Method				Covariance Method				Convex Relaxation			
Rank (n)	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.
2	0.42233	6.820%	0.04528	0.731%	0.46859	7.568%	0.06908	1.116%	0.34606	5.589%	0.11423	1.845%
3	0.21459	3.610%	0.07782	1.309%	0.30045	5.054%	0.06946	1.168%	0.12264	2.063%	0.03302	0.555%
4	0.37109	6.362%	0.07639	1.310%	1.53712	26.354%	7.95890	136.458%	0.10961	1.879%	0.01257	0.216%

In the fully-determined case, the spectral methods perform reasonably well in terms of the MNE. Once again, the covariance method performs poorly with recovery up to rank-4, however, errors are below 8% for the MNE. The ENE is also very favourable, with errors under 2%, with the exception of the covariance method at rank-4. In terms of convex relaxation, performance is better than the spectral methods in the MNE, and actually exhibits improved accuracy as the recovery is performed to higher ranks (as expected). The ENE is also excellent and more consistent, with it dropping as low 0.22% at rank-4.

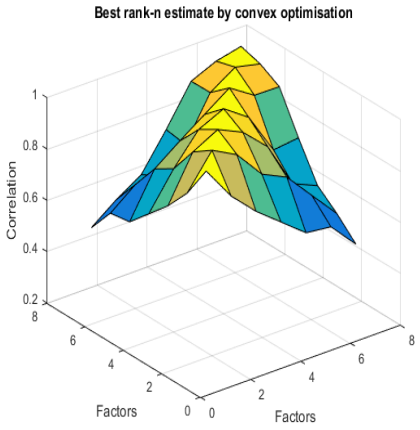
Tab. 5.4: Under-determined noisy Toeplitz system comparison of completion methods.

Toeplitz	Correlation Method				Covariance Method				Convex Relaxation			
Rank (n)	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.
3	0.31426	5.286%	0.21975	3.696%	1.91784	32.261%	0.65960	11.095%	0.26007	4.375%	0.18684	3.143%
4	0.44321	7.599%	0.12726	2.182%	0.89692	15.378%	0.26691	4.576%	0.26189	4.490%	0.14523	2.490%
5	0.45215	7.839%	0.20166	3.496%	0.84946	14.727%	8.06640	139.845%	0.26426	4.581%	0.11167	1.936%

Due to the presence of noise in this parametrisation, it was expected that under-determined testing would be a challenge. Indeed, the covariance method does not perform very well, across all ranks. Somewhat surprisingly, however, the correlation method still performs reasonably well with MNEs of $< 8\%$ and ENEs of $< 4\%$. In comparison, convex relaxation, once again, performs significantly better than the spectral methods. MNEs are consistently $< 5\%$, while ENEs hover between 1.9-3.2%. The ENE actually tends to improve with increasing rank. Surface plots are given in Figure 5.4.

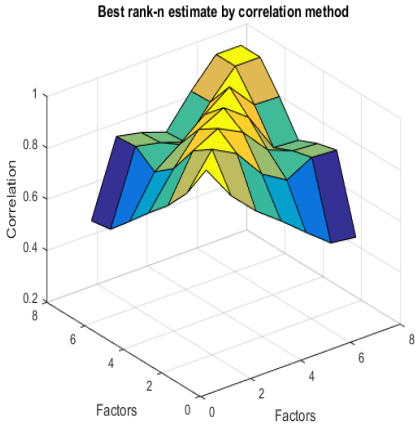


(a) Using $M = 3$, $n = 4$

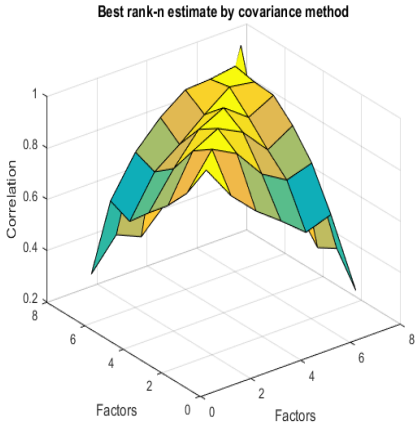


(b) Convex relaxation method, $n = 4$.

Fig. 5.3: Comparison of true low-rank matrix and convex relaxation approximation for an under-determined noisy Toeplitz system.



(a) Correlation method, $n = 4$.



(b) Covariance method, $n = 4$.

Fig. 5.4: Comparison of correlation and covariance approximations for an under-determined noisy Toeplitz system.

As can be seen, the convex relaxation method most closely resembles the true, low-rank matrix, while the spectral methods could not achieve a good fit.

5.2 Real Data Testing

Testing involving parametric correlation structures are very important, as they provide a way in which to test the accuracy of the algorithms. They are an indicator of how well the algorithms perform and serve to highlight any shortcomings in the methodology. The true test, however, lies in the application of these techniques to real-world data.

5.2.1 FX Rates Data

The first dataset consisted of FX rates data for 250 business days between 01/2017 and 12/2017. These included the USD/ZAR, USD/AUD, USD/CAD, USD/EUR, USD/GBP and USD/JPY exchange rates. From the time-series rates data, a sample correlation matrix was generated. In order to enforce some degree of structure, the rates were arranged into a crude pseudo-Toeplitz form (for reasons explained later), whereby correlations were larger nearer the main diagonal, and decreased towards the boundaries. The full-rank, true FX rates correlation matrix is given in Figure 5.5.

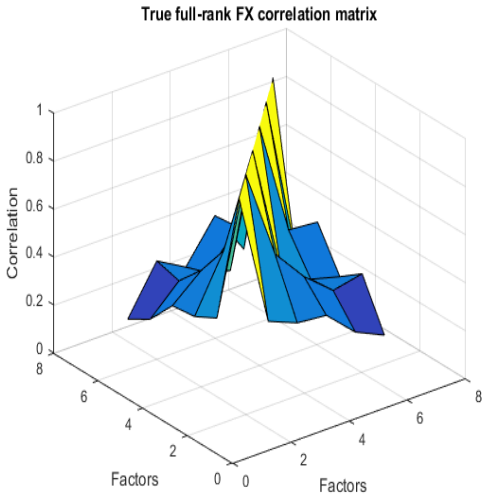


Fig. 5.5: Full-rank FX rates correlation matrix.

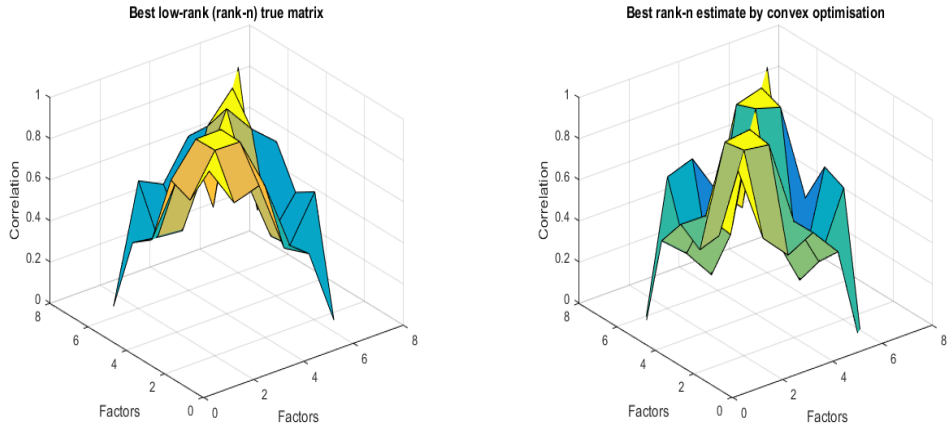
For the purposes of testing, a scenario was chosen whereby it was assumed that the first four FX rates were known ($N = 4$) and the last two rates were added in as additional factors to the system ($d = 2$), namely the USD/GBP and USD/JPY. A set of correlation pairs was then selected, and these were used as inputs to each of the completion algorithms (see Appendix D.2). Recovery to different ranks was again performed and each of the completion methods were compared. The results are summarised in Table 5.5.

Tab. 5.5: Comparison of completion methods for real FX rates data.

FX Rates	Correlation Method				Covariance Method				Convex Relaxation			
Rank (n)	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.
2	2.02711	41.647%	0.45546	9.358%	2.21134	45.432%	0.70025	14.387%	0.96055	19.735%	0.37648	7.735%
3	1.80272	41.862%	0.26953	6.259%	1.67708	38.945%	0.30250	7.024%	0.94095	21.850%	0.58692	13.629%
4	1.43372	36.984%	0.31009	7.999%	1.40858	36.336%	5.61020	144.722%	1.63571	42.195%	1.01239	26.116%

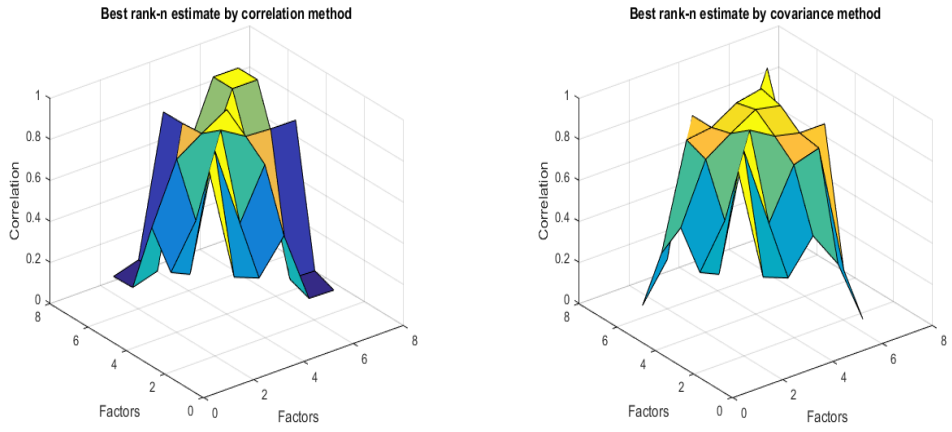
As expected, the completion methods perform inconsistently, as there is now no definitive underlying structure on which to rely. MNEs are, in general, quite high, however, convex relaxation is about 50% more accurate than the spectral methods. The ENE results are interesting, as the correlation method consistently yields the lowest errors, except for rank-2, where convex relaxation is the best at $< 8\%$. The

covariance method is the least consistent of the three, performing especially poorly at rank-4. An interesting observation is that even though convex relaxation did not perform as well as expected, the approximation most closely fits the true, low-rank matrix (as can be seen in Figures 5.6 and 5.7). This is, indeed, a very promising result, as it indicates that convex relaxation is still approximately mirroring the underlying behaviour of the system. This is in contrast to the spectral methods, where the fit seems arbitrary and there is very little resemblance (even though the error characteristics between all three methods are comparable).



(a) True, low-rank matrix, $n = 3$. **(b)** Convex relaxation approximation, $n = 3$.

Fig. 5.6: Comparison of true low-rank matrix and convex relaxation approximation for real FX rates data.



(a) Correlation approximation, $n = 3$. **(b)** Covariance approximation, $n = 3$.

Fig. 5.7: Comparison of correlation and covariance approximations for real FX rates data.

To illustrate the problem in a more descriptive manner, consider the partially observed FX rates correlation matrix given as follows:

$$\mathbf{M} = \begin{array}{c} \\ \$/R \\ \$/A\$ \\ \$/C\$ \\ \$/€ \\ \$/£ \\ \$/¥ \end{array} \begin{array}{c} \begin{array}{cccccc} \$/R & \$/A\$ & \$/C\$ & \$/€ & \$/£ & \$/¥ \end{array} \\ \begin{pmatrix} 1.000 & 0.528 & 0.403 & 0.373 & 0.439 & 0.039 \\ 0.528 & 1.000 & 0.983 & 0.575 & ? & ? \\ 0.403 & 0.983 & 1.000 & 0.454 & ? & 0.567 \\ 0.373 & 0.575 & 0.454 & 1.000 & 0.863 & ? \\ 0.439 & ? & ? & 0.863 & 1.000 & 0.372 \\ 0.039 & ? & 0.567 & ? & 0.372 & 1.000 \end{pmatrix} \end{array}.$$

Note that the 4×4 top-left sub-matrix is itself of low-rank (rank-3), as per the general problem setup. **C**, below, represents the FX rates correlation matrix post-completion (note the original user-defined low-rank matrix is preserved). The overall rank, however, is some value greater than 3, which is the rank we require:

$$\mathbf{C} = \begin{array}{c} \\ \$/R \\ \$/A\$ \\ \$/C\$ \\ \$/€ \\ \$/£ \\ \$/¥ \end{array} \begin{array}{c} \begin{array}{cccccc} \$/R & \$/A\$ & \$/C\$ & \$/€ & \$/£ & \$/¥ \end{array} \\ \begin{pmatrix} 1.000 & 0.528 & 0.403 & 0.373 & 0.439 & 0.039 \\ 0.528 & 1.000 & 0.983 & 0.575 & 0.306 & 0.536 \\ 0.403 & 0.983 & 1.000 & 0.454 & 0.157 & 0.567 \\ 0.373 & 0.575 & 0.454 & 1.000 & 0.863 & 0.288 \\ 0.439 & 0.306 & 0.157 & 0.863 & 1.000 & 0.372 \\ 0.039 & 0.536 & 0.567 & 0.288 & 0.372 & 1.000 \end{pmatrix} \end{array}.$$

The final approximated matrix is then obtained by applying modified PCA, to rank-reduce **C** to rank-3.

5.2.2 Stock Price Data

The second dataset consisted of daily time-series stock price data for the Top 10 JSE-listed stocks (period: 01/2017 - 01/2019), by market capitalisation. They are shown in Table 5.6, with the average 2-year stock prices given.

Tab. 5.6: Average stock prices for the Top 10 JSE-listed stocks (01/2017-01/2019).

Company	JSE-Code	Avg. Stock Price (R)
Naspers	NPN	2976.28
British American Tabacco	BAC	763.38
Glencore	GLN	58.56
BHP Billiton	BHP	255.95
Richemont	CFR	111.25
Anglo American	AGL	259.98
FirstRand	FSR	59.28
Standard Bank	SBK	175.53
Sasol	SOL	432.62
Vodacom	VOD	147.82

From this data, a 10×10 sample correlation matrix was generated (using the full time-series). For the purposes of testing, the scenario created assumed that the user provided the correlations between 7 stocks ($N = 7$), with 3 additional stocks being added to the system ($d = 3$). Further, only 9 correlation inputs were provided, in an attempt to perform an under-determined, worst-case type testing (see Appendix D.2). Once again, in order to enforce some degree of structure to the data, the correlations were arranged into a crude pseudo-Toeplitz form (as best as possible). The full-rank stock correlation matrix is depicted in Figure 5.8.

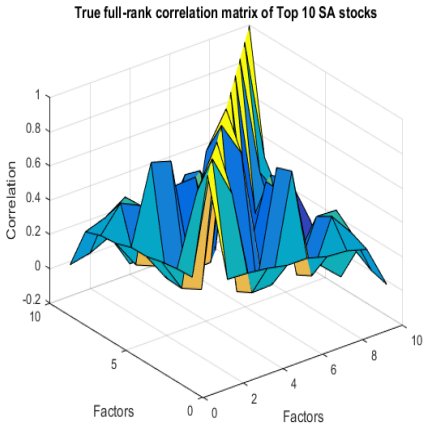


Fig. 5.8: Full-rank Top 10 JSE stocks correlation matrix

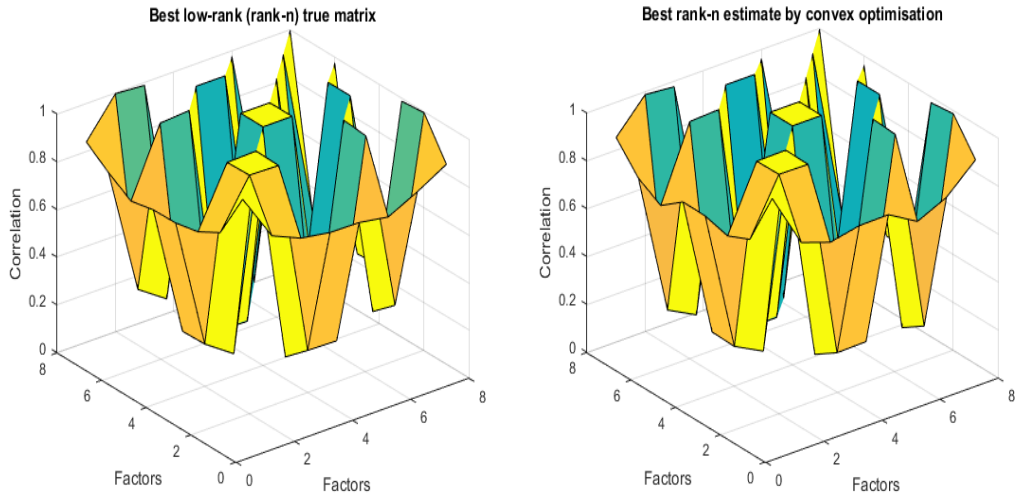
Table 5.7 summarises the testing results for completion by each of the different methods.

Tab. 5.7: Comparison of completion methods for real stock data.

Stocks	Correlation Method				Covariance Method				Convex Relaxation			
Rank (n)	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.
2	7.80847	108.600%	0.18481	2.570%	0.31836	4.428%	0.06930	0.964%	0.27231	3.787%	0.03555	0.494%
3	5.34607	84.909%	0.26816	4.259%	5.29672	84.126%	0.39236	6.232%	2.42082	38.449%	1.40263	22.277%
4	2.89619	50.537%	1.26775	22.122%	4.14315	72.296%	1.28574	22.435%	2.45162	42.779%	1.12829	19.688%

As can be seen, both the correlation and covariance methods yield extremely large errors and experience significant difficulty with the completion. The MNEs are very large, while the ENEs, in contrast, are relatively good. The covariance method in particular performs surprisingly well for a rank-2 recovery, with an MNE of $< 5\%$ and ENE of $< 1\%$. When convex relaxation is applied, the results tell a different story. It performs better than the spectral methods by a large margin, being especially effective at a rank-2 level. Further, due to the system being heavily under-determined, it was expected that errors would increase rapidly as we attempted recovery to higher ranks. However, at rank-2, convex relaxation yields an MNE of $< 4\%$ and ENE of $< 0.5\%$, which is remarkable. In fact, when considering the fitting of the approximation to the true, low-rank matrix, and repeating the process with only 6 correlation inputs instead of 9, the results are remarkable. Consider the surface plots seen in Figure 5.9.

From the plots, we note an almost exact fit to the true, low-rank rank-2 stock price correlation matrix. What makes this result very encouraging, is that it is achieved by only providing 6 correlation inputs to the convex relaxation algorithm; the 3 boundary correlations (see Appendix D.2) and 3 intra-correlations (between the additional stocks). As per the trend observed thus far, the spectral methods result in erratic fitting, with minimal resemblance to the true, low-rank matrix.



(a) True, low-rank matrix, $n = 2$. **(b)** Convex relaxation approximation, $n = 2$.

Fig. 5.9: Comparison of true low-rank matrix and convex relaxation approximation for a rank-2 stock price correlation matrix with 6 input pairs given.

To illustrate the scenario in a more descriptive manner, consider the partially observed stock price correlation matrix given as follows:

$$\mathbf{M} = \begin{array}{c} \\ NPN \\ AGL \\ GLN \\ FSR \\ SBK \\ BHP \\ VOD \\ SOL \\ CFR \\ BAC \end{array} \begin{array}{c} \begin{array}{cccccccccc} NPN & AGL & GLN & FSR & SBK & BHP & VOD & SOL & CFR & BAC \end{array} \\ \begin{pmatrix} 1.000 & 0.768 & 0.725 & 0.744 & 0.746 & 0.681 & 0.731 & 0.838 & 0.573 & -0.066 \\ 0.768 & 1.000 & 0.998 & 0.144 & 0.147 & 0.992 & 0.124 & ? & ? & ? \\ 0.725 & 0.998 & 1.000 & 0.080 & 0.083 & 0.998 & 0.060 & ? & ? & ? \\ 0.744 & 0.144 & 0.080 & 1.000 & 1.000 & 0.018 & 1.000 & ? & ? & ? \\ 0.746 & 0.147 & 0.083 & 1.000 & 1.000 & 0.021 & 1.000 & ? & ? & ? \\ 0.681 & 0.992 & 0.998 & 0.018 & 0.021 & 1.000 & -0.002 & ? & ? & ? \\ 0.731 & 0.124 & 0.060 & 1.000 & 1.000 & -0.002 & 1.000 & 0.295 & ? & ? \\ 0.838 & ? & ? & ? & ? & ? & 0.295 & 1.000 & 0.927 & ? \\ 0.573 & ? & ? & ? & ? & ? & ? & 0.927 & 1.000 & 0.780 \\ -0.066 & ? & ? & ? & ? & ? & ? & ? & 0.780 & 1.000 \end{pmatrix} \end{array}.$$

Note that the 7×7 top-left sub-matrix is itself of low-rank (rank-2), as per the general problem setup. Matrix $\mathbf{C}$, on the following page, represents the stock price correlation matrix post-completion (note the original user-defined low-rank matrix is preserved). The overall rank, however, is some value greater than 2, which is the rank we require:

$$\mathbf{C} = \begin{array}{c} \\ NPN \\ AGL \\ GLN \\ FSR \\ SBK \\ BHP \\ VOD \\ SOL \\ CFR \\ BAC \end{array} \begin{array}{c} \begin{array}{cccccccccc} NPN & AGL & GLN & FSR & SBK & BHP & VOD & SOL & CFR & BAC \end{array} \\ \begin{pmatrix} 1.000 & 0.768 & 0.725 & 0.744 & 0.746 & 0.681 & 0.731 & 0.838 & 0.573 & -0.066 \\ 0.768 & 1.000 & 0.998 & 0.144 & 0.147 & 0.992 & 0.124 & 0.939 & 0.881 & 0.484 \\ 0.725 & 0.998 & 1.000 & 0.080 & 0.083 & 0.998 & 0.060 & 0.926 & 0.889 & 0.527 \\ 0.744 & 0.144 & 0.080 & 1.000 & 1.000 & 0.018 & 1.000 & 0.314 & -0.036 & -0.610 \\ 0.746 & 0.147 & 0.083 & 1.000 & 1.000 & 0.021 & 1.000 & 0.316 & -0.033 & -0.608 \\ 0.681 & 0.992 & 0.998 & 0.018 & 0.021 & 1.000 & -0.002 & 0.909 & 0.894 & 0.566 \\ 0.731 & 0.124 & 0.060 & 1.000 & 1.000 & -0.002 & 1.000 & 0.295 & -0.055 & -0.623 \\ 0.838 & 0.939 & 0.926 & 0.314 & 0.316 & 0.909 & 0.295 & 1.000 & 0.927 & 0.489 \\ 0.573 & 0.881 & 0.889 & -0.036 & -0.033 & 0.894 & -0.055 & 0.927 & 1.000 & 0.780 \\ -0.066 & 0.484 & 0.527 & -0.610 & -0.608 & 0.566 & -0.623 & 0.489 & 0.780 & 1.000 \end{pmatrix} \end{array}.$$

After applying modified PCA, the best low-rank rank-2 estimate is given as follows:

$$\hat{\mathbf{C}} = \begin{array}{c} \\ NPN \\ AGL \\ GLN \\ FSR \\ SBK \\ BHP \\ VOD \\ SOL \\ CFR \\ BAC \end{array} \begin{array}{c} \begin{array}{cccccccccc} NPN & AGL & GLN & FSR & SBK & BHP & VOD & SOL & CFR & BAC \end{array} \\ \begin{pmatrix} 1.000 & 0.772 & 0.729 & 0.755 & 0.757 & 0.685 & 0.742 & 0.854 & 0.607 & -0.050 \\ 0.772 & 1.000 & 0.998 & 0.166 & 0.169 & 0.992 & 0.146 & 0.990 & 0.974 & 0.596 \\ 0.729 & 0.998 & 1.000 & 0.102 & 0.105 & 0.998 & 0.082 & 0.979 & 0.986 & 0.647 \\ 0.755 & 0.166 & 0.102 & 1.000 & 1.000 & 0.039 & 1.000 & 0.304 & -0.063 & -0.693 \\ 0.757 & 0.169 & 0.105 & 1.000 & 1.000 & 0.042 & 1.000 & 0.307 & -0.060 & -0.690 \\ 0.685 & 0.992 & 0.998 & 0.039 & 0.042 & 1.000 & 0.019 & 0.964 & 0.995 & 0.694 \\ 0.742 & 0.146 & 0.082 & 1.000 & 1.000 & 0.019 & 1.000 & 0.285 & -0.083 & -0.707 \\ 0.854 & 0.990 & 0.979 & 0.304 & 0.307 & 0.964 & 0.285 & 1.000 & 0.931 & 0.476 \\ 0.607 & 0.974 & 0.986 & -0.063 & -0.060 & 0.995 & -0.083 & 0.931 & 1.000 & 0.764 \\ -0.050 & 0.596 & 0.647 & -0.693 & -0.690 & 0.694 & -0.707 & 0.476 & 0.764 & 1.000 \end{pmatrix} \end{array}.$$

The true, rank-2 matrix is given in Appendix D for reference.

5.2.3 Testing the Effect of Varying the Number of Input Correlations

The final and very important facet of the numerical testing, involved examining the effect of changing the number of input correlations to the algorithms. This directly ties into part of the research question we have attempted to address thus far, whereby we aim to determine which correlation inputs are the most important for completion and the minimum number required for reasonable results. The testing was performed on the real stock data (under the same scenario) using the convex relaxation method, as it was clearly the most accurate and robust of the techniques. Testing was performed for 6, 9, 12 and 15 correlation inputs, with recovery done to different ranks as before. Note, the results for 6 correlations were discussed briefly in Section 5.2.2 and the surface plots are shown in Figure 5.9. The results for 9 correlations can be found in Table 5.7. For a full list of the input correlations used in testing, refer to Appendix D.2.

As mentioned earlier, the 6 correlation inputs consisted of the boundary correlations, as well as the intra-correlations between the 3 additional stocks (i.e. VOD/SOL, SOL/CFR and CFR/BAC). Note that these correlations were used as inputs from the true, low-rank matrix $\mathbf{S}$, given in Appendix D. Thereafter, inter-correlations between the original N factors and additional d factors were chosen at random, along columns 8, 9 and 10, which served as the additional correlation inputs. It should

Tab. 5.8: Comparison of completion results for different combinations of input correlations.

Convex Relaxation	6 Inputs Given				12 Inputs Given				15 Inputs Given			
Rank (n)	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.	X_F	% Diff.	X_{EV}	% Diff.
2	0.35321	4.912%	0.05593	0.778%	0.27969	3.890%	0.04208	0.585%	0.27015	3.757%	0.05027	0.699%
3	2.93670	46.642%	1.42535	22.638%	1.21143	19.241%	0.31977	5.079%	0.71860	11.413%	0.04043	0.642%
4	2.70220	47.152%	1.08991	19.018%	2.18374	38.105%	0.86877	15.160%	0.53410	9.320%	0.12459	2.174%

be further noted that there are a total of 24 additional correlations that are added to the system, due to the addition of the extra 3 factors (SOL, CFR and BAC).

In Table 5.8, we see the comparisons for 6, 12 and 15 (out of 24 unknown) correlation inputs given. The first general observation is that for each of the ranks, as more inputs are given, the errors become smaller. For the rank-3 case, the MNE drops from 46.6% to 11.4% and the ENE from 22.6% to 0.6%, as the number of given inputs increases from 6 to 15. The same behaviour is noted for recoveries to rank-4 and rank-2, although the results are less pronounced in the case of the latter. There is, in fact, a very slight increase of approximately 0.1% in the ENE from 12 to 15 correlations, at rank-2.

Another important observation is that the lower the required rank, the fewer input correlations from the recovered matrix are required. In other words, if we endeavour to recover to higher rank, we require more inputs to be given. With only 6 inputs, an MNE of 4.9% and ENE of 0.8%, is achieved, for a rank-2 recovery. These errors jump to over 40% and 20% respectively, if we attempt recovery to ranks 3 and 4, while still only providing 6 inputs. Similarly, it can be seen that for 12 and 15 given inputs, we can achieve the best recoveries to ranks 3 and 4 respectively.

In summary, a minimum of 6 inputs are required for reasonable recovery to rank-2. For rank-3, a minimum of 12 inputs yields a highly respectable ENE of $< 5.1\%$, although the MNE is still slightly high. If we target recovery to rank-4, a minimum of 15 inputs are required, which results in an MNE of $< 10\%$ and ENE of $< 2.2\%$, which is excellent. Naturally, it may be assumed that this trend will continue, as more inputs are given.

5.3 Discussion

The results obtained from numerical testing have been, generally, quite favourable. All three completion methods performed well in certain instances. The spectral methods tended to perform better with the parametric correlation structures, and better MNEs and ENEs were achieved for fully-determined systems. When the systems were under-determined, the performance of the spectral methods tended to decrease, save for a few exceptions. In almost every instance, convex relaxation produced better results than the spectral methods, and proved to be especially robust in the under-determined test-cases.

A noteworthy point is that even though the MNE and ENE values for the spectral methods are, at times, comparable with those of convex relaxation, the fitting achieved by the latter was always far superior to those achieved by the former. This

is evident in the surface plots, where the greatest resemblance was always between the convex relaxation approximated matrices and the true, low-rank matrices. This inconsistency of fitting by the spectral methods may lie in the fact that completion was achieved by fitting to discrete point masses, as opposed to accounting for the general distribution of energy in the matrices. Instead of fitting by a closeness argument, as in the case of convex relaxation, the spectral methods pinned the mass down to individual correlation inputs, thereby lacking control over the missing correlations. This was one of the major limitations of this technique, which detracted from its overall robustness and accuracy.

In terms of real data testing, the primary challenge lay in the inherent lack of structure in these correlation matrices. It was found that by enforcing a very crude pseudo-Toeplitz structuring (which is not always possible) on the given data, the completion was more accurate and the fitting much improved. The effect of the nature of the underlying structure and the degree to which it affects completion, is an area that warrants further investigation. The type of data may also play a role, as we see the performance of completion for stock price data being better than that for FX rates. Another pointed observation, which may explain the challenges with real data, lies in the principal components and method of rank-reduction used. Recalling that modified PCA was applied, which is based on an eigendecomposition and truncation of eigenvalues, the accuracy of the results decreases, as the magnitude of the truncated eigenvalues increases. As noted in Section 2.3.1, most of the variation in a dataset is generally recovered within the first few eigenvalues, with the remainder becoming negligibly small. This, therefore, allows truncation to be effective, as not much information is being lost. However, with real data this is not always the case; in particular, the eigenvalues do not necessarily diminish and are no longer negligibly small (as was particularly noted in the FX rates data). Therefore, modified PCA was not the most appropriate technique to have used for the rank-reduction, as non-negligible amounts of information was being lost from the system. To improve this, an alternative technique of rank-reduction should be explored (e.g. majorisation), which may not alter the PCs of the system as crudely. This may lead to better results and overall performance.

The next important discussion point revolves around the choice of preserving the original user-defined $N \times N$ rank-n matrix or modifying it, as we search for the best approximation to the true, optimal low-rank structure. If the user requires that the original $N \times N$ matrix is preserved exactly, with an overall system rank of n, then it is only the spectral methods that can achieve this. However, as we have seen, these methods generally result in non-optimality. Nonetheless, if this is what the user desires, perhaps due to the advantages of not having to adjust the pricing routines based on the $N \times N$ system, then the user must be content to sacrifice overall fit and accuracy in favour of these advantages. On the other hand, if the user desires to incorporate the additional factors in the most consistent and accurate way possible, and is willing to allow the original $N \times N$ matrix to be adjusted (albeit slightly) in favour of greater accuracy, then the mantle falls upon convex relaxation. This method has demonstrated the best fitting and closest approximations to the true, low-rank matrices, and is therefore, the best choice for obtaining the most consistent and robust low-rank solutions overall.

In terms of which inputs are the most important, it is proposed that the boundary correlations are the most critical, followed by the intra-correlations between the additional factors. Thereafter, at least one other correlation should be provided along each of the columns (and corresponding rows due to symmetry) of the additional factors. This should be a correlation which lies half-way along the unknown entries of each column (and corresponding row), or alternatively, if extra inputs can be determined, they should be chosen so that they fill out the unknown entries in a regular manner (along each column and corresponding row). It was also pleasing to observe the general reduction in errors, as more input correlations were provided to the algorithms. Intuitively, this improvement in the approximations was natural to assume, due to there being less unknown information in the system. By increasing the number of input correlations, the system may become over-determined and the spectral methods will require a least-squares solution involving the Moore-Penrose pseudoinverse (see Section 2.3.2). This, however, will still yield non-optimal results when compared to convex relaxation, which only improves the more over-determined the system becomes.

Chapter 6

Extensions and Conclusions

Whilst the approaches taken in this dissertation have yielded favourable results, there are a few notable areas of extension to the work. The first would be to consider improved and more sophisticated techniques for finding optimal low-rank approximations to correlation matrices. Modified PCA, whilst simple and effective, has the noted disadvantage of non-optimality. More sophisticated techniques, in particular majorisation, can be explored, which would improve the overall accuracy of the low-rank approximations. Indeed, this would be much more favourable when working with real data, for the reasons outlined in Section 5.3.

Another area to consider are mixed-type or hybrid correlation structures. These would be correlation matrices that contain correlations between different classes of instruments, for example interest rates and stocks. Consider a user who has a 3-factor LMM, and intends to incorporate a 2-factor stock model. In order to preserve the overall rank-5 structure, these cross-correlations between the interest rates and stocks need to somehow be determined, most likely from a partially observed matrix. This would necessitate the application of some form of matrix completion.

The next area involves further investigation into the effects and nature of the underlying structure (if any) for real data, and its effects on completion. Exploring methods of characterising this structure and perhaps enforcing a more consistent form of structuring (instead of the highly crude pseudo-Toeplitz form used), may yield better approximations. Results indicated that structure does play a role, with better performance being noted when a degree of structure was enforced on the data (albeit crudely).

Yet another perspective to consider is that, often, correlations cannot simply be estimated from historical data. Many instruments in the market are forward-looking, and as such, there exists a forward-looking correlation structure, not unlike a forward-looking volatility matrix. Hence, data simply does not exist from which to construct these correlations, which drives the need to explore methods of low-rank matrix completion under these scenarios.

As we have seen, completion by convex relaxation was performed directly in terms of correlation structures. An alternative method to consider, akin to what was done in the spectral methods, is to perform the convex relaxation using covariances instead, and then backing out the correlation structures. This would remove some of the constraints in the solution space and simplify the objective function. A further method, involves the application of non-convex optimisation techniques (as outlined in Section 2.5.4), as they have been shown to be more effective for

large-scale problems. In addition, the application of collaborative filtering and so-called restricted Boltzmann machines (Koren, 2009), which were more advanced techniques used in the Netflix Prize, warrant further investigation. A comparison of results from these various methods would make for interesting analysis.

In terms of the work presented thus far, the methods of spectral completion and convex relaxation were successfully applied to the problem of low-rank completion and recovery of correlation matrices. The spectral methods relied on the use of the spectral decomposition of a real, symmetric matrix and comprised of two sub-techniques based on correlation and covariance formulations respectively. In contrast, convex relaxation involved a direct low-rank completion by relaxing the rank-minimisation heuristic, and reformulating the problem in terms of a nuclear norm minimisation (Candès and Recht, 2009).

Two correlation matrix parametrisations were successfully used for testing, namely the classical exponential form of Brigo and Mercurio (2006) and the noisy Toeplitz parametrisation of Hardin *et al.* (2013). The aforementioned completion techniques were first tested on the above parametric forms, before being tested on real datasets comprising of FX rates and stock price data. Both fully and under-determined testing was performed, and in almost all cases, the convex relaxation method performed significantly better than the spectral methods. Even in instances where MNEs and ENEs were comparable between the methods, convex relaxation always resulted in the closest fit to the optimal, low-rank matrices.

In terms of real data testing, convex relaxation proved to be the superior technique, once again. Even with the challenges associated with real data, including the lack of apparent structure and distribution of PCs, convex relaxation remained the most consistent method. It was deemed that the type and nature of the data has an effect on the low-rank recovery, with this being noted as an area warranting further investigation.

The effects of varying the number of given correlation inputs was also successfully tested. The accuracy of the low-rank approximations was found to be directly proportional to the number of given inputs. Testing was performed on stock data, with MNEs and ENEs dropping from maximums of 47.1% and 22.6% to minimums of 9.3% and 0.6% respectively (as the number of input correlations increased from 6 to 15). Further, it was determined that the lower the required rank to which recovery is aimed, the fewer the number of inputs required. Recalling the research question (as stated in Chapter 1) which this work has attempted to address, the following additional conclusions can be drawn. If rank-recovery for the system is required to rank-r, then the minimum number of correlation pairs required as input is r, per column (and corresponding row). A bare minimum of one input pair should be given, however, r-many would theoretically yield the most reliable results (see Section 2.5.3). In terms of which input correlation pairs are the most important, it was found that the boundary correlations are the most crucial, followed by the intra-correlations between the additional factors (which lie adjacent to the main diagonal). In general, if more correlations along the unknown column (and row) spaces can be determined (to fill out the space in a regular manner) and provided as inputs, then more reliable recovery is possible.

In terms of the choice of preserving the original user-defined $N \times N$ rank-n

matrix or allowing it to be modified, the choice is dependent on the requirements of the user. If it is required that the original $N \times N$ system is strictly preserved, there will be a corresponding sacrifice in accuracy and overall fit, as only the spectral methods can achieve this. If, however, the user allows the original $N \times N$ system to be modified slightly, then convex relaxation can be applied and greater accuracy in the low-rank approximation of the overall system will be achieved.

As a final point, and in addition to the work heretofore presented, perhaps the true power of low-rank correlation matrix completion and the techniques explored thus far, lies in their generality. Correlation structures exist and can be constructed for countless applications and phenomena around us, of which their utility within the sphere of finance, is but one example. Thus, this work itself, in the broadest sense, can be easily extended to a plethora of areas, due to the transferability and effectiveness of these techniques. This, indeed, is an exciting prospect.

Bibliography

Bektaş, S. and Şişman, Y. (2010). The comparison of L_1 and L_2-norm minimization methods, *International Journal of Physical Sciences* **5**(11): 1721–1727.

Brigo, D. and Mercurio, F. (2006). *Interest Rate Models - Theory and Practice*, 2 edn, Springer.

Candès, E. J. and Plan, Y. (2010). Matrix completion with noise, *Proceedings of the IEEE* **98**(6): 925–936.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization, *Foundations of Computational Mathematics* **9**(6): 717.

Candès, E. J. and Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion, *IEEE Transactions on Information Theory* **56**(5): 2053–2080.

Caveny, J. L. and Murray, J. F. (1995). Data Reduction: An Introduction to Principal Component Analysis, *U.S. Quality Algorithms, Inc.* pp. 781–790.

Chi, Y. (2018). Low-Rank Matrix Completion, *IEEE Signal Processing Magazine* . Available: https://users.ece.cmu.edu/~yuejiec/papers/SPM_lrmc_final.pdf.

Fazel, M. (2002). *Matrix rank minimization with applications*, PhD thesis, Stanford University.

Hardin, J., Garcia, S. R. and Golan, D. (2013). A method for generating realistic correlation matrices, *The Annals of Applied Statistics* pp. 1733–1762.

Jiang, Y. and Lyu, S. (2017). Simple introduction to matrix completion, *Olin College of Engineering - Linearity1 – Final Project* . Available at: https://drive.google.com/file/d/0BxX6w2-tbbg8eVZfck5lc253WHM/view.

Karamizadeh, S., Abdullah, S. M., Manaf, A. A., Zamani, M. and Hooman, A. (2013). An Overview of Principal Component Analysis, *Journal of Signal and Information Processing* **4**: 173–175.

Koren, Y. (2009). The Bellkor Solution to the Netflix Grand Prize, *Netflix prize documentation* **81**: 1–10.

MacAusland, R. (2014). The Moore-Penrose Inverse and Least Squares, *Math 420: Advanced Topics in Linear Algebra* pp. 1–10.

Martin, C. D. and Porter, M. A. (2012). The Extraordinary SVD, *The American Mathematical Monthly* **119**(10): 838–851.

McWalter, T. (2018). Tutorial 7 - Figures and Plots, *Mathematical Computing Skills* .

Pham, L. (2013). Principal Component Analysis and Singular Value Decomposition in Matrix Dimensionality Reduction and Covariance/Correlation Estimation, *L-Stern Group* . Available at:
https://talksonmarkets.files.wordpress.com/2012/09/principal-component-analysis-e28093-covariance1.pdf.

Pietersz, R. and Groenen, P. J. (2004). Rank reduction of correlation matrices by majorization, *Quantitative Finance* **4**(6): 649–662.

Rebonato, R. and Jäckel, P. (1999). The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. Available at SSRN: https://ssrn.com/abstract=1969689.

Zhang, Z. and Wu, L. (2003). Optimal low-rank approximation to a correlation matrix, *Linear Algebra and its Applications* **364**: 161–187.

Appendix A

Method of PCA

The following algorithm outlines the process of PCA, as presented in McWalter (2018).

Consider an $m \times n$ matrix $\mathbf{X}$ of correlated observations, where each column is a set of observations. Then,

1. Compute the mean of the m random variables as the sum over the elements in each row divided by n.
2. Subtract the mean vector from each column of observations in $\mathbf{X}$ so that there is no residual mean.
3. Compute the covariance matrix $\mathbf{C} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\top$.
4. Compute the eigenvectors (and store as columns in a matrix $\mathbf{W}$) and eigenvalues of $\mathbf{C}$.
5. The matrix $\mathbf{W}$ is the orthogonal transform which may be used to transform $\mathbf{X}$ into its principle components $\mathbf{Y} = \mathbf{W}^\top\mathbf{X}$.

This procedure may be applied to raw datasets of observations to recover the covariance (and subsequently correlation) structure.

Appendix B

Non-Convex Optimisation Procedure

Recall the the modified loss function given in Section 2.5.4 (Chi, 2018):

$$F(\mathbf{X}, \mathbf{Y}) = \frac{1}{4p} f(\mathbf{X}, \mathbf{Y}) + \frac{1}{16} \|\mathbf{X}^\top \mathbf{X} - \mathbf{Y}^\top \mathbf{Y}\|_F^2,$$

where p is the observation probability and the second term is included to account for solutions where $\mathbf{X}$ and $\mathbf{Y}$ have balanced norms.

Chi (2018) provides the following optimisation procedure for the non-convex loss function $F(\mathbf{X}, \mathbf{Y})$:

1. The first step applies the so-called spectral method, to find an initialisation that is close to the ground truth factorisation of $\mathbf{M}$. Consider the partially observed matrix $\frac{1}{p} P_\Omega(\mathbf{M})$, which is an unbiased estimate of $\mathbf{M}$ with expectation $\mathbb{E}[\frac{1}{p} P_\Omega(\mathbf{M})] = \mathbf{M}$. Hence, a reasonably good initial guess is produced from its best rank-r approximation. With application of the SVD, let $\mathbf{U}_0 \mathbf{\Sigma}_0 \mathbf{V}_0^\top$ be the best rank-r approximation of $\frac{1}{p} P_\Omega(\mathbf{M})$, where $\mathbf{U}_0 \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{V}_0 \in \mathbb{R}^{n_2 \times r}$ are orthonormal and $\mathbf{\Sigma}_0$ is an $r \times r$ diagonal matrix. Thus, $\mathbf{X}_0 = \mathbf{U}_0 \mathbf{\Sigma}_0^{1/2}$ and $\mathbf{Y}_0 = \mathbf{V}_0 \mathbf{\Sigma}_0^{1/2}$, via the spectral initialisation.

2. Next, the initial estimate is locally refined via simple iterative methods (such as gradient descent), following the update rule,

$$\begin{bmatrix} \mathbf{X}_{t+1} \\ \mathbf{Y}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_t \\ \mathbf{Y}_t \end{bmatrix} - \eta_t \begin{bmatrix} \nabla_{\mathbf{X}} F(\mathbf{X}_t, \mathbf{Y}_t) \\ \nabla_{\mathbf{Y}} F(\mathbf{X}_t, \mathbf{Y}_t) \end{bmatrix},$$

   where η_t is the step size, and $\nabla_{\mathbf{X}} F(\mathbf{X}_t, \mathbf{Y}_t)$, $\nabla_{\mathbf{Y}} F(\mathbf{X}_t, \mathbf{Y}_t)$ are partial derivatives with respect to $\mathbf{X}$ and $\mathbf{Y}$ respectively.

We now apply the SVD to $\mathbf{M}$ (as per (2.4)) and define the following quantities, $\mathbf{X}^\star = \mathbf{U} \mathbf{\Sigma}^{1/2}$ and $\mathbf{Y}^\star = \mathbf{V} \mathbf{\Sigma}^{1/2}$. Hence, the factorisation of $\mathbf{M}$ can then be written as $\mathbf{M} = \mathbf{X}^\star \mathbf{Y}^{\star\top}$, and call $\mathbf{Z}^\star = [\mathbf{X}^{\star\top}, \mathbf{Y}^{\star\top}]^\top \in \mathbb{R}^{(n_1+n_2) \times r}$ the ground truth. Now, define the optimal transform between the tth iterate $\mathbf{Z}_t = [\mathbf{X}_t{}^\top, \mathbf{Y}_t{}^\top]^\top \in \mathbb{R}^{(n_1+n_2) \times r}$ and $\mathbf{Z}^\star$ as

$$\underset{\mathbf{R} \in \mathbb{R}^{r \times r},\ \mathbf{R}\mathbf{R}^\top = \mathbf{I}}{\operatorname{argmin}} \|\mathbf{Z}_t \mathbf{R} - \mathbf{Z}^\star\|_F,$$

since $\mathbf{Z}^\star$ is only identifiable up to orthonormal transforms. Assume the condition number $\kappa := \frac{\sigma_1}{\sigma_r}$ of $\mathbf{M}$ is a bounded constant, then as long as,

$$p \geq C_1 \frac{\mu^3 r^3 \log^3 n}{n},$$

for some sufficiently large constant $C_1 > 0$, with high probability, the iterates satisfy,

$$\|\mathbf{Z}_t \mathbf{H}_t - \mathbf{Z}^\star\|_F \leq C_2 \rho^t \mu r \frac{1}{\sqrt{np}} \|\mathbf{Z}^\star\|_F \quad \forall t \geq 0,$$

where $C_2 > 0$, $0 < \rho < 1$ are some constants (provided that the step-size $0 < \eta_t \equiv \eta \leq \frac{2}{25\kappa\sigma_1}$). Therefore, once the number of measurements is of the order $\mu^3 r^3 n \log^3 n$, gradient descent converges at a geometric rate. Note that this scales linearly in n (up to logarithmic factors). If we now define $\frac{\|\mathbf{Z}_t \mathbf{H}_t - \mathbf{Z}^\star\|_F}{\|\mathbf{Z}^\star\|_F} < \epsilon$ as the ϵ-level accuracy, then gradient descent requires an order of $\log 1/\epsilon$ iterations. It should be further noted that the number of iterations is independent of the problem-size, which results in a lower cost per iteration and lower computational cost overall.

Appendix C

Proof of the Convex Envelope of Rank Theorem (2.12)

We start with the *Convex Envelope of Rank* theorem, as presented in Section 2.5.3 (Fazel, 2002). The *convex envelope* of $f : \mathcal{C} \to \mathbb{R}$ is defined as the largest convex function g, such that $g(x) \leq f(x)$ for all $x \in \mathcal{C}$, where $\mathcal{C} \subseteq \mathbb{R}^n$. In other words, g is the closest (point-wise) to f, amongst all convex functions.

Theorem C.1 (Convex Envelope of Rank)**.** *Consider an $m \times n$ matrix $\boldsymbol{X}$. On the set $\mathcal{S} = \{\boldsymbol{X} \in \mathbb{R}^{m \times n} : \|\boldsymbol{X}\| \leq 1\}$, the convex envelope of the function $\phi(\boldsymbol{X}) = \text{rank}(\boldsymbol{X})$ is $\phi_{\text{env}}(\boldsymbol{X}) = \|\boldsymbol{X}\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(\boldsymbol{X})$, where σ_i are the singular-values of $\boldsymbol{X}$.*

The proof is given as follows (adapted directly from Fazel (2002)):

We begin by defining the notion of *conjugate functions*. The conjugate f^* of a function $f : \mathcal{C} \to \mathbb{R}$, where $\mathcal{C} \subseteq \mathbb{R}^n$, is defined as,

$$f^*(y) = \sup\{\langle y, x\rangle - f(x) \mid x \in C\},$$

where $\langle y, x\rangle$ denotes the inner product in $\mathbb{R}^n$, and $C \in \mathcal{C}$. A further basic result of convex analysis is that the convex envelope of a function f, is the conjugate of the conjugate, given as f^{**}.

Part a). *Computing ϕ^**: On the set of matrices with spectral norm ≤ 1, the conjugate of the rank function ϕ, is given by,

$$\phi^*(Y) = \sup_{\|X\| \leq 1} (\mathbf{Tr}(Y^\top X) - \phi(X)), \tag{C.1}$$

where $\langle Y, X\rangle = \mathbf{Tr}(Y^\top X)$ is the inner product in $\mathbb{R}^{m \times n}$. Let $q = \min\{m, n\}$. By von Neumann's trace theorem,

$$\mathbf{Tr}(Y^\top X) \leq \sum_{i=1}^{q} \sigma_i(Y)\sigma_i(X), \tag{C.2}$$

where $\sigma_i(X)$ denotes the ith largest singular value of X. If given Y, and if U_X and V_X are chosen to be equal to U_Y and V_Y respectively, then equality in (C.2) will be achieved. Here, $X = U_X \Sigma_X V_X^\top$ and $Y = U_Y \Sigma_Y V_Y^\top$ are the SVDs of X and Y. The

term $\phi(X)$ in (C.1) is independent of U_X and V_X, hence to find the supremum, we set $U_X = U_Y$ and $V_X = V_Y$, in order to maximise the first term. It follows that,

$$\phi^*(Y) = \sup_{\|X\| \leq 1} \left(\sum_{i=1}^{q} \sigma_i(Y)\sigma_i(X) - \text{rank}(X) \right).$$

If $X = 0$, we have $\phi^*(Y) = 0$ for all Y. If $\text{rank}(X) = r,\ 1 \leq r \leq q$, then $\phi^*(Y) = \sum_{i=1}^{r} \sigma_i(Y) - r$. Hence, $\phi^*(Y)$ can be expressed as,

$$\phi^*(Y) = \max \left\{ 0,\ \sigma_1(Y) - 1, \ldots, \sum_{i=1}^{r} \sigma_i(Y) - r, \ldots, \sum_{i=1}^{q} \sigma_i(Y) - q \right\}.$$

The largest term in this set is the one that sums all *positive* $(\sigma_i(Y) - 1)$ terms. It can be concluded that,

$$\begin{aligned} \phi^*(Y) &= \begin{cases} 0 & \text{for } \|Y\| \leq 1, \\ \sum_{i=1}^{r} (\sigma_i(Y) - r) & \text{for } \sigma_r(Y) > 1 \text{ and } \sigma_{r+1}(Y) \leq 1. \end{cases} \\ &= \sum_{i=1}^{q} (\sigma_i(Y) - 1)_+, \end{aligned}$$

where $a_+ = \max\{0, a\}$.

Part b). *Computing* ϕ^{**}: We now find the conjugate of ϕ^*, defined as,

$$\phi^{**}(Z) = \sup_Y (\mathbf{Tr}(Z^\top Y) - \phi^*(Y)),$$

for all $Z \in \mathcal{C}^{m \times n}$. As before, let $U_Y = U_Z$ and $V_Y = V_Z$ to get,

$$\phi^{**}(Z) = \sup_Y \left(\sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \phi^*(Y) \right).$$

Consider two cases, $\|Z\| > 1$ and $\|Z\| \leq 1$: If $\|Z\| > 1$, we can choose $\sigma_1(Y)$ large enough so that $\phi^{**}(Z) \to \infty$. This is evidenced by the fact that in,

$$\phi^{**}(Z) = \sup_Y \left(\sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \left(\sum_{i=1}^{r} \sigma_i(Y) - r \right) \right),$$

the coefficient of $\sigma_1(Y)$ is $(\sigma_1(Z) - 1)$ which is positive. Now, we let $\|Z\| \leq 1$. If $\|Y\| \leq 1$, then $\phi^*(Y) = 0$ and the supremum is achieved for $\sigma_i(Y) = 1,\ i = 1, \ldots, q$, yielding,

$$\phi^{**}(Z) = \sum_{i=1}^{q} \sigma_i(Z) = \|Z\|_*.$$

The next important step is to show that the argument of the *sup* is always $<$ the value given above, if $\|Y\| > 1$.

With an addition and subtraction of the $\sum_{i=1}^{q} \sigma_i(Z)$ term and rearranging, we have,

$$\begin{aligned}
&\sum_{i=1}^{q} \sigma_i(Y)\sigma_i(Z) - \sum_{i=1}^{r}(\sigma_i(Y) - 1) \\
&= \sum_{i=1}^{q} \sigma_i(Y)\sigma_i(Z) - \sum_{i=1}^{r}(\sigma_i(Y) - 1) - \sum_{i=1}^{q} \sigma_i(Z) + \sum_{i=1}^{q} \sigma_i(Z) \\
&= \sum_{i=1}^{r}(\sigma_i(Y) - 1)(\sigma_i(Z) - 1) + \sum_{i=r+1}^{q} (\sigma_i(Y) - 1)\sigma_i(Z) + \sum_{i=1}^{q} \sigma_i(Z) \qquad \text{(C.3)} \\
&< \sum_{i=1}^{q} \sigma_i(Z).
\end{aligned}$$

Note that the first two terms in (C.3) always have a negative value, which results in the final inequality holding.

In summary, we have therefore shown,

$$\phi^{**}(Z) = \|Z\|_*,$$

over the set $\{Z \mid \|Z\| \leq 1\}$. Thus, over this set, $\|Z\|_*$ is the convex envelope of the function rank(Z), thus concluding the proof.

Appendix D

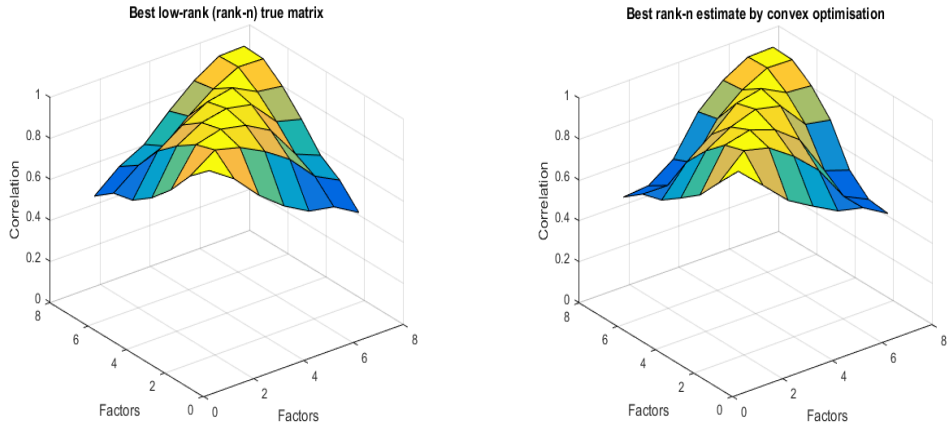
Additional Results and Testing Details

D.1 Additional Results

True, rank-2 stock correlation matrix:

$$\mathbf{S} = \begin{array}{c} \\ NPN \\ AGL \\ GLN \\ FSR \\ SBK \\ BHP \\ VOD \\ SOL \\ CFR \\ BAC \end{array} \begin{array}{c} \begin{array}{cccccccccc} NPN & AGL & GLN & FSR & SBK & BHP & VOD & SOL & CFR & BAC \end{array} \\ \begin{pmatrix} 1.000 & 0.803 & 0.749 & 0.729 & 0.727 & 0.706 & 0.768 & 0.839 & 0.571 & -0.076 \\ 0.803 & 1.000 & 0.996 & 0.178 & 0.174 & 0.989 & 0.235 & 0.998 & 0.948 & 0.533 \\ 0.749 & 0.996 & 1.000 & 0.093 & 0.090 & 0.998 & 0.151 & 0.989 & 0.972 & 0.603 \\ 0.729 & 0.178 & 0.093 & 1.000 & 1.000 & 0.030 & 0.998 & 0.239 & -0.145 & -0.738 \\ 0.727 & 0.174 & 0.090 & 1.000 & 1.000 & 0.027 & 0.998 & 0.236 & -0.149 & -0.740 \\ 0.706 & 0.989 & 0.998 & 0.030 & 0.027 & 1.000 & 0.089 & 0.978 & 0.985 & 0.652 \\ 0.768 & 0.235 & 0.151 & 0.998 & 0.998 & 0.089 & 1.000 & 0.296 & -0.087 & -0.697 \\ 0.839 & 0.998 & 0.989 & 0.239 & 0.236 & 0.978 & 0.296 & 1.000 & 0.926 & 0.479 \\ 0.571 & 0.948 & 0.972 & -0.145 & -0.149 & 0.985 & -0.087 & 0.926 & 1.000 & 0.775 \\ -0.076 & 0.533 & 0.603 & -0.738 & -0.740 & 0.652 & -0.697 & 0.479 & 0.775 & 1.000 \end{pmatrix} \end{array}.$$

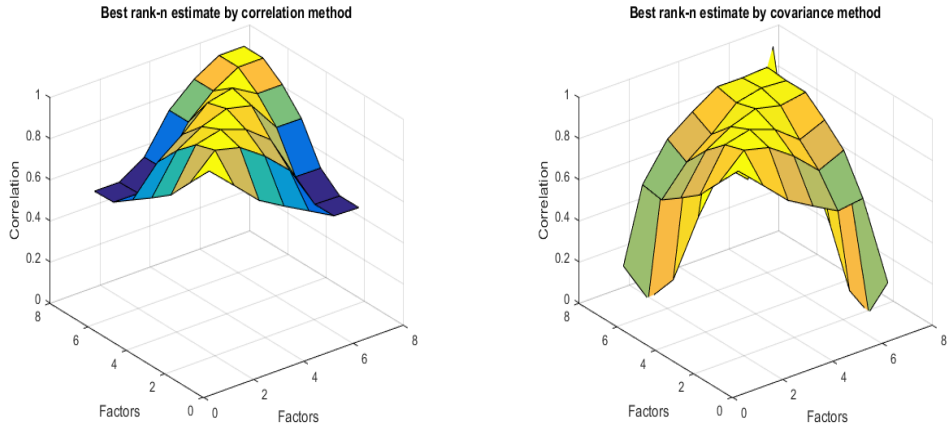
D.1.1 Further Plots



(a) Using $M = 3$, $n = 3$.

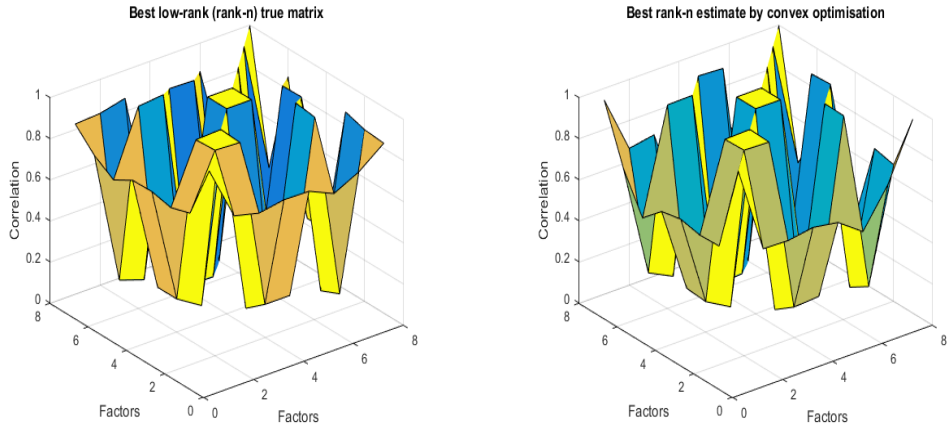
(b) Convex relaxation, $n = 3$.

Fig. D.1: Comparison of true low-rank matrix and convex relaxation approximation for an under-determined noisy Toeplitz matrix.



(a) Correlation method, $n = 3$. **(b)** Covariance method, $n = 3$.

Fig. D.2: Comparison of correlation and covariance approximations for an underdetermined noisy Toeplitz matrix.



(a) True, low-rank matrix, $n = 3$. **(b)** Convex relaxation approximation, $n = 3$.

Fig. D.3: Comparison of true low-rank matrix and convex relaxation approximation for a rank-3 stock price correlation matrix with 12 correlation inputs.

D.2 Testing Details

The following is a list of the correlations provided as inputs to each of the test cases outlined in Chapter 5. Note for a correlation (a, b) given below, take the transpose (b, a) as automatically included as well, due to the symmetry in the correlation structure.

1. **Fully-determined testing**: $N = 5$, $d = 2$
   (a) Rank 2 - (1,6), (1,7), (6,7), (4,7).
   (b) Rank 3 - (1,6), (1,7), (6,7), (3,7), (3,6), (5,7).

(c) Rank 4 - (1,6), (1,7), (6,7), (3,7), (3,6), (5,7), (5,6), (4,7).

2. **Under-determined testing**: $N = 5$, $d = 2$

   (a) For all ranks – (1,6), (1,7), (5,6), (4,7), (6,7).

3. **FX rates testing**: $N = 4$, $d = 2$

   (a) For all ranks – (1,5), (1,6), (4,5), (3,6), (5,6).

4. **Stock data testing**: $N = 7$, $d = 3$

   (a) 6 Inputs – (1,8), (1,9), (1,10), (7,8), (8,9), (9,10).

   (b) 9 Inputs – (1,8), (1,9), (1,10), (7,8), (8,9), (9,10), (4,8), (4,9), (5,10).

   (c) 12 Inputs – (1,8), (1,9), (1,10), (7,8), (8,9), (9,10), (4,8), (5,9), (4,10), (3,9), (3,10), (8,10).

   (d) 15 Inputs – (1,8), (1,9), (1,10), (7,8), (8,9), (9,10), (4,8), (5,9), (4,10), (3,9), (3,10), (8,10), (7,10), (6,8), (6,9).