Efficient PSD Matrix Approximation by Iterative Optimisations and Gradient Descent Method

Runshi Wang

Bayes Business School, City, University of London, UK

Vali Asimit

Bayes Business School, City, University of London, UK

Rui Zhu

Bayes Business School, City, University of London, UK

Feng Zhou

University of Cambridge, UK and Bayes Business School, City, University of London, UK

Abstract

We devise two algorithms for approximating solutions of PSDisation, a problem in actuarial science and finance, to find the nearest valid correlation matrix that is positive semidefinite. The first method converts the PSDisation problem with a positive semidefinite constraint and other linear constraints into iterative linear programmings (LP) or quadratic programmings (QP). The LPs or QPs in our formulation give an upper bound of the optimal solution of the original problem which can be improved during each iteration. The biggest advantage of this iterative method is its great flexibility when working with different choices of norms or with user-defined constraints. Secondly, a gradient descent method is designed specifically for PSDisation under the Frobenius norm to measure how close the two metrices are. Experiments on randomly generated data show that this method enjoys better resilience to noise while maintaining good accuracy. Examples of applications in finance as well as machine learning field are given. Computational results are presented followed by discussion on future improvements.

Keywords: Nearest correlation matrix; Positive semidefinite; Semidefinite programming.

1. Introduction

A correlation matrix is a square matrix summarising correlation coefficients between each pair of variables. This is used by many financial or insurance companies to determine how a group of risks are dependent from each other (Pearson, 1895). A major usage of correlation matrix is to measure an insurer's exposure to risks and calculate the Solvency Capital Requirement (SCR) under the Solvency II Standard Formula (Milhaud et al., 2018). Another application for the correlation matrix is to work with copulas for aggregation of risks in more complex capital

Email addresses: runshi.wang@bayes.city.ac.uk (Runshi Wang), asimit@city.ac.uk (Vali Asimit), rui.zhu@city.ac.uk (Rui Zhu), feng.zhou@city.ac.uk and feng.zhou@mrc-bsu.cam.ac.uk (Feng Zhou)

model (Milhaud et al., 2018).

Both applications require the correlation matrix to be symmetric and positive semidefinite with diagonal entries all equal to one. In practice, however, the correlation matrix estimated from the empirical data is rarely positive semidefinite due to a variety of reasons, including data incompleteness, noise, rounding, manual adjustment or inconsistent computing approaches, etc (Cutajar et al., 2017). Therefore, a valid correlation matrix that is positive semidefinite should be obtained based on the empirical matrix in order to perform risk analysis. The problem of finding such correlation matrix nearest to an empirical matrix is called PSDisation (Milhaud et al., 2018).

Denote the set of $n \times n$ real matrices by $\mathbb{R}^{n\times n}$. We write $\mathbf{Z} \succeq 0$ if $\mathbf{Z}$ is a symmetric positive semidefinite matrix ($\mathbf{Z} \succ 0$ for positive definite). We use $\mathrm{diag}(\mathbf{Z})$ to denote vector of all diagonal entries of $\mathbf{Z}$ and $\mathbf{1}$ stands for the all-ones vector. The operator $||\cdot||$ stands for some matrix norm, for which the choice will be explained later.

The PSDisation problem we consider in this paper is defined as follows. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix that is not positive semidefinite with diagonal entries all equal to 1. It represents an empirical correlation matrix of n random variables $Y_1, \ldots, Y_n$ with standard deviations $\sigma_{Y_1}, \ldots, \sigma_{Y_n}$ whose (i, j) entry is given by

$$a_{ij} = \mathrm{corr}(Y_i, Y_j) = \frac{\mathrm{cov}(Y_i, Y_j)}{\sigma_{Y_i}\sigma_{Y_j}}, \text{ if } \sigma_{Y_i}\sigma_{Y_j} > 0.$$

We aim to find the nearest positive semidefinite matrix $\mathbf{X}$ of the same form. The problem can be written as follows.

Problem 1.

$$\begin{aligned} \min_{\mathbf{X}} \ & ||\mathbf{A} - \mathbf{X}||^2 \\ s.t. \ & \mathbf{X} \succeq 0, \\ & \mathrm{diag}(\mathbf{X}) = \mathbf{1}, \end{aligned} \tag{1.1}$$

As we wish to find the nearest correlation matrix $\mathbf{X}$ to the empirical matrix $\mathbf{A}$, the choice of the matrix norm $||\cdot||$ determines how "nearest" is defined. In this paper, we suggest four different norms to choose from.

A trivial choice is the Frobenius norm, or the F-norm (Van Loan and Golub, 1983),

$$||\mathbf{M}||_{\mathbf{F}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |m_{ij}|^2} = \sqrt{\mathrm{Tr}(\mathbf{M}^{\mathbf{T}}\mathbf{M})}.$$

Another well-known norm is the W-norm (Higham, 2002),

$$||\mathbf{M}||_{\mathbf{W}} = ||\mathbf{W}^{\frac{1}{2}}\mathbf{M}\mathbf{W}^{\frac{1}{2}}||_{\mathbf{F}},$$

where $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a square matrix with positive entries. This weighted norm is commonly used in numerical mathematics (Higham, 2002) and allows us to force some elements of $\mathbf{X}$ to be closer to the corresponding entries in $\mathbf{A}$. Setting $\mathbf{W}$ to the identity matrix will retrieve the F-norm.

However, when applied to actuarial science, an insurance company may be more interested in the H-norm (Milhaud et al., 2018) defined by

$$||\mathbf{M}||_{\mathbf{H}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} h_{ij}|m_{ij}|^2}.$$

This is similar to the F-norm but allows one to assign weights to $\mathbf{A}$ on an element-by-element basis. Therefore the H-norm is preferred when one has prior knowledge on the correlation matrix $\mathbf{A}$, *e.g.* how confident one is in each correlation coefficient. It is equivalent to the F-norm when $\mathbf{H}$ is an all-ones matrix.

Finally, we have the Chebyshev norm, or the max norm, as an alternative, which calculates the maximum difference between each entry in $\mathbf{A}$ and its corresponding element in $\mathbf{X}$ (Cantrell, 2000):

$$||\mathbf{M}||_{\mathbf{max}} = \max_{i,j}(|m_{ij}|).$$

The Chebyshev norm specifies an element-wise ceiling for the largest difference between entries of $\mathbf{X}$ and $\mathbf{A}$. This can be particularly useful and more robust to outliers when m_{ij} is large.

Currently most state-of-the-art methods focus on finding the nearest correlation matrix with respect to the F-norm, of which some apply to H-norm as well. Over those algorithms, the Newton method (NM) (Qi and Sun, 2006) and the augmented Lagrangian method (ALD) (Qi and Sun, 2011) are the most efficient in producing optimal solutions. The Newton method is able to work with W-norm and ALD is designed for H-norm, where upper bound and lower bound for each entry of the matrix can also be set. The alternating projections method (APM) aims to find the nearest PSD matrix by iteratively projecting $\mathbf{A}$ onto two convex sets (Higham, 2002). It converges linearly at best (Deutsch and Hundal, 1997) and can be applied under the F-norm as well as the W-norm. The spectral decomposition method (SDM) is one of the easiest way to tackle the PSDisation problem. By setting all negative eigenvalues to zero after performing a spectral decomposition of the initial matrix followed by a rescaling, it can give an acceptable approximation of the solution to the PSDisation problem even faster. The shirking method proposed by Higham et al. (2016) has been designed to handle correlation matrix with fixed diagonal blocks. However, the outcome of this method heavily depends on the choice of a target matrix which is positive semidefinite and near to $\mathbf{A}$. Therefore it is not considered suitable for finding the global optimal of the PSDisation problem. Cutajar et al. (2017) explored the shrinking method as a way to improve the results of the alternating projections method. There are other PSDisation methods such as semidefinite programming (Higham,

2002) and hypersphere decomposition method (Rebonato and Jäckel, 2011). For complete survey and a full list of PSDisation algorithms please see Cutajar et al. (2017); Milhaud et al. (2018).

However, depending on the industrial situations, insurance companies may wish to measure the similarity using an alternative norm other than the F-norm, which makes some algorithms more time-consuming or invalid. In this paper, we propose two new methods to approximate the solution of the PSDisation problem. The first algorithm works by solving a series of optimisations such as linear programmings (LP) or quadratic programmings (QP). Instead of solving the original problem which requires the correlation matrix $\mathbf{X}$ to be positive semidefinite, our formulation requires $\mathbf{X} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$ where $\mathbf{Q}$ belongs to the convex cone of diagonally dominant symmetric matrices with non-negative diagonal entries and thus all constraints are linear. This is a rich subset of the PSD matrices set, but the optimisations can be solved more efficiently due to the linearity of the constraints. The main advantage of this method is its flexibility as our formulation is adapted directly from the original problem and is thus effective on any choice of norms, including the F-norm, the Chebyshev norm and the H-norm. The advantages of using these norms will be illustrated in our experiment results. The second algorithm is to find the nearest correlation matrix to the initial matrix using an iterative gradient projection method under the F-norm. This can be done simply by repeatedly taking a step along the gradient of the objective function and then projecting the matrix back to the positive semidefinite cone. This algorithm is easy to implement and enjoys good efficiency while maintaining good accuracy.

Extensive experiments demonstrate that our first iterative algorithm could achieve comparable results to state-of-art methods, while showing more flexibility to handle with complex constraints or different choices of norms. Our second gradient descent method provides slightly less accurate results, but runs faster than the first algorithm and is more resilient to noise than some state-of-art methods and therefore can be more reliable in practical situations.

This paper is organised as follows. In Section 2 we introduce our two approaches to the PSDisation problem. Experiment results for both algorithms are reported in Section 3. Finally, in Section 4 we conclude this paper and propose some future work.

2. Proposed PSDisation Methods

In this section, two methods are proposed to find the nearest correlation matrix. Currently most PSDisation methods are designed specifically for the F-norm. Some of them can be adapted to work with the H-norm with a cost of more complex formulation or higher computational time. Our first proposed method is done by iteratively solving a series of linearly constrained quadratic optimisations or linear optimisations. Therefore, we get an approximation of the optimal value of the PSDisation problem by directly minimising the objective function at each iteration, where we have the flexibility to easily choose different norms to work with. The other approach is based on iteratively taking a step along the gradient to reduce the objective function and projecting onto the positive semidefinite set with diagonal one according to spectral decomposition method (Rebonato and Jäckel, 2011), which is designed for solving the PSDisation problem with respect to the F-norm.

2.1. Iterative Quadratic/Linear Programming

Consider the PSDisation problem in (1.1), which requires $\mathbf{X}$ to be positive semidefinite in its constraints. The positive semidefinite constraint can be reduced and therefore the optimisation problem is transfered into quadratic or linear optimisation problems with linear constraints.

A square matrix $\mathbf{X}$ is called diagonally dominant if

$$|x_{ii}| \geq \sum_{j \neq i} |x_{ij}| \;\; \forall \; 1 \leq i \leq n.$$

It is well-known that a symmetric diagonally dominant matrix with non-negative diagonal entries is positive semidefinite, which makes up a rich subset of PSD matrices. Therefore, we could approximate the optimisation problem (1.1) stated above by solving convex optimisation problems of the following form in an iterative manner,

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{Q}} \; ||\mathbf{A} - \mathbf{X}||^2 \\ &s.t. \;\; \text{diag}(\mathbf{X}) = \mathbf{1}, \\ &\quad\;\; \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\ &\quad\;\; \mathbf{Q} = \mathbf{Q}^T, \\ &\quad\;\; \mathbf{Q} \text{ is diagonally dominant with non-negative diagonal.} \end{aligned} \tag{2.1}$$

Note that $\mathbf{U}_1 = \mathbf{I}_d$ and $\mathbf{U}_h = \text{Chol}(\mathbf{X}^*_{h-1})$ for $h \geq 2$ where $\mathbf{X}^*_{h-1}$ is the optimal solution from the previous optimisation. $\mathbf{U} = \text{Chol}(\mathbf{X})$ represents the Cholesky decomposition of $\mathbf{X}$ such that $\mathbf{U}$ is an upper triangular matrix satisfying $\mathbf{X} = \mathbf{U}^T\mathbf{U}$.

Lemma 2. *Problem* (2.1) *gives an upper bound of Problem* (1.1)*.*

Proof. The matrix $\mathbf{Q}$ is guaranteed to be positive semidefinite in (2.1), since it is symmetric and diagonally dominant with non-negative diagonal entries and therefore $\mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h$ is also positive semidefinite. Problem (2.1) minimises the objective function over a subset of positive semidefinite matrices and therefore Lemma 2 holds. □

Lemma 3. *Problem* (2.1) *is feasible for iteration h ($h \geq 2$) if it is feasible for iteration $h-1$.*

Proof. This proof is trivial as $\mathbf{X}^*_{h-1} = \mathbf{U}_h^T \mathbf{I} \mathbf{U}_h$ and the identity matrix $\mathbf{I}$ is symmetric and diagonally dominant. Therefore, solution of problem in iteration $h-1$ is feasible for iteration h. □

Let us denote the objective function by $f(\cdot)$,

$$f(\mathbf{X}) = ||\mathbf{A} - \mathbf{X}||^2.$$

Lemma 2 and Lemma 3 indicate that an optimal solution always exists for (2.1) as long as the problem in the first problem is feasible and the optimal solution in each iteration is at least as good as that from the previous iteration, *i.e.,*

$$f(\mathbf{X}_h^*) \leq f(\mathbf{X}_{h-1}^*) \quad \forall h \geq 2.$$

Given that the problems are feasible and the optimal solution is positive definite (which is true when numerical computation is concerned), it can be further shown that the optimal value of Problem (2.1) decrease strictly after each iteration unless it reaches the optimal value of Problem (1.1) (Ahmadi and Hall, 2017). In this case, the optimal objective value in each iteration in Problem (2.1) will finally converge as it is monotonic decreasing and bounded below by the true optimal value of Problem (1.1). In our numerical experiments, the optimal after each iteration always finally converges to the true optimal and strong empirical evidence shows a fast convergence.

Algorithm 1: An iterative quadratic/linear programming algorithm to solve PSDisation under some matrix norm

```
$\mathbf{X}_0 = \mathbf{I}$;
while not converged do
    $\mathbf{U}_h = \mathrm{Chol}(\mathbf{X}_{h-1})$;
    find $\mathbf{X}_h$ by solving
        $\min_{\mathbf{X}_h, \mathbf{Q}_h} ||\mathbf{A} - \mathbf{X}_h||^2$
        $s.t.\ \mathrm{diag}(\mathbf{X}_h) = \mathbf{1}$,
            $\mathbf{X}_h = \mathbf{U}_h^T \mathbf{Q}_h \mathbf{U}_h$,
            $\mathbf{Q}_h = \mathbf{Q}_h^T$,
            $\mathbf{Q}_h$ is diagonally dominant with non-negative diagonal;
    $h = h + 1$;
end
```

Let us take the F-norm as an example. Note that $||\mathbf{A} - \mathbf{X}||_{\mathbf{F}}^2 = \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{a}^T\boldsymbol{x} + \boldsymbol{a}^T\boldsymbol{a}$ where $\boldsymbol{x} = \mathrm{vec}(\mathbf{X})$ and $\boldsymbol{a} = \mathrm{vec}(\mathbf{A})$ and vec is the *vectorisation function* that transforms a matrix into a column vector. We further rewrite the objective function and the constraint that $\mathbf{Q}$ is diagonally dominant with non-negative diagonal by adding slack variables $\mathbf{R}$ into the optimisation,

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{Q}, \mathbf{R}} \ & \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{a}^T\boldsymbol{x} \\ s.t.\ & \mathrm{diag}(\mathbf{X}) = \mathbf{1}, \\ & q_{ii} \geq \sum_{j \neq i} r_{ij}, \qquad 1 \leq i \leq n \end{aligned}$$

$$
\begin{aligned}
& -r_{ij} \leq q_{ij} \leq r_{ij}, && 1 \leq i \neq j \leq n \\
& \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\
& \mathbf{Q} = \mathbf{Q}^T.
\end{aligned}
$$

Now all constraints become linear and we obtained a series of iterative QPs. Each QP has $3n^2$ variables and $4n^2 - n$ constraints, which can be solved efficiently.

Similarly, if the H-norm is applied in the objective function, the iterative QPs can be written as

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{Q},\mathbf{R}} \ & \frac{1}{2} \boldsymbol{x}^T \operatorname{Diag}(\boldsymbol{h}) \boldsymbol{x} - \boldsymbol{a}^T \operatorname{Diag}(\boldsymbol{h}) \boldsymbol{x} \\
s.t. \ & \operatorname{diag}(\mathbf{X}) = \mathbf{1}, \\
& q_{ii} \geq \sum_{j \neq i} r_{ij}, && 1 \leq i \leq n \\
& -r_{ij} \leq q_{ij} \leq r_{ij}, && 1 \leq i \neq j \leq n \\
& \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\
& \mathbf{Q} = \mathbf{Q}^T.
\end{aligned}
$$

where $\boldsymbol{h} = \operatorname{vec}(\mathbf{H})$ and $\operatorname{Diag}(\boldsymbol{h})$ is an $n^2 \times n^2$ diagonal matrix with the elements of $\boldsymbol{h}$ on the main diagonal. This general formulation makes our method more flexible when an insurance company wants to assign weight on each entry of the correlation matrix, exploiting prior knowledge on the data.

Next, we consider the situation where we optimise with respect to the Chebyshev norm. According to (Cutajar et al., 2017), PSDisation over the F-norm is likely to result in a correlation matrix $\mathbf{X}$ in which some entries differ significantly from the initial matrix $\mathbf{A}$, while other entries have relatively smaller deviations, giving the minimum F-norm optimal. However, this is not preferred by some insurance companies as they aim to minimise the maximum discrepancy between corresponding entries from the valid correlation matrix $\mathbf{X}$ and initial matrix $\mathbf{A}$. In this case, minimising over the Chebyshev norm becomes an alternative choice. With our iterative approximating method, this can be obtained by solving a series of LPs of the following form,

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{Q},\mathbf{R},t} \ & t \\
s.t. \ & -t \leq a_{ij} - x_{ij} \leq t, && 1 \leq i \neq j \leq n \\
& \operatorname{diag}(\mathbf{X}) = \mathbf{1}, \\
& q_{ii} \geq \sum_{j \neq i} r_{ij}, && 1 \leq i \leq n \\
& -r_{ij} \leq q_{ij} \leq r_{ij}, && 1 \leq i \neq j \leq n \\
& \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\
& \mathbf{Q} = \mathbf{Q}^T.
\end{aligned}
$$

Experiments show that our iterative algorithm proposed for approximating optimal solutions for PSDisation problems offers great flexibility under different norms while maintaining a good accuracy. Details are explained in Section 3.

2.2. Gradient Descent

Next we propose our second algorithm, where we will focus on working under the F-norm only as there is currently no known method that could project $\mathbf{X}$ onto the positive semidefinite set under the H-norm. This method is therefore less flexible in the choice of norms. However, it works more efficiently and provides more steady results on real data with noise.

Consider the objective of the PSDisation problem in (1.1),

$$\min_{\mathbf{X}} ||\mathbf{A} - \mathbf{X}||^2,$$

the gradient under the F-norm is calculated as

$$\frac{\partial}{\partial \mathbf{X}} ||\mathbf{A} - \mathbf{X}||_{\mathbf{F}} = \frac{\mathbf{X} - \mathbf{A}}{||\mathbf{A} - \mathbf{X}||_{\mathbf{F}}}.$$

To project a matrix $\mathbf{X}$ onto the positive semidefinite cone $\mathcal{S}_{+}^{n}$, first perform the eigendecomposition such that $\mathbf{X} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$ and $\mathbf{\Lambda} = \mathrm{Diag}(\lambda_i)$ where λ_i are eigenvalues of $\mathbf{X}$. Define $\mathbf{\Lambda}_{+} = \mathrm{Diag}(\lambda_{i+})$ where $\lambda_{i+} = \max(\lambda_i, 0)$. Next, in order to ensure that the resulting matrix has diagonal $\mathbf{1}$, we calculate the scaling matrix $\mathbf{T} = \mathrm{Diag}(t_i)$ where the weighting parameter t_i is given by

$$t_i = \sqrt{(\sum_{m=1}^{n} q_{im}^2 \lambda_{m+})^{-1}}.$$

Then $\mathbf{X}_{+} = \mathbf{T}\mathbf{Q}\mathbf{\Lambda}_{+}\mathbf{Q}^{-1}\mathbf{T}$ is the projection of $\mathbf{X}$ onto the cone of positive semidefinite matrices.

Our proposed method is shown in Algorithm 2 below. $\mathbf{X}$ is updated iteratively by taking a gradient step and projection until converged. According to our experiments, the gradient step γ can be initialised as a large value to speedup the convergence and it will fast decrease during iterations.

Algorithm 2: The Gradient descent algorithm for PSDisation under the F-norm

```
$\mathbf{X}_0 = \mathbf{I}$;
while not converged do
    $\mathbf{X}_{i+1} = \mathbf{X}_i - \gamma \mathbf{G}_i$;
    $\mathbf{X}_{i+1} = \mathcal{P}_{\mathcal{S}^+}(\mathbf{X}_{i+1})$;
    $C_{i+1} = ||\mathbf{A} - \mathbf{X}_{i+1}||_{\mathbf{F}}$;
    if $C_{i+1} < C_i$ then
        $\gamma = \gamma * (1 + \delta)$;
    else
        $\gamma = \gamma / 2$;
    end
    $i = i + 1$;
end
```

3. Experimental Results

To illustrate the robustness and flexibility of our algorithms, we designed a series of tests based on simulated and real datasets. In this section, we show the experimental results and compare our methods with the state-of-the-arts.

3.1. Experiments on Iterative Quadratic/Linear Programming Algorithm

To test our iterative quadratic/linear programming algorithm, we randomly generate the initial matrix **A** of different dimensions (*i.e.* with dimensions of 5, 10, 25, 50 and 75) that is not positive semidefinite. For **A** generated from each dimension settings, we solve for the nearest valid correlation matrix **X** using our method for the F-norm and the Chebyshev norm, which lead to iterative quadratic programming (IQP) and iterative linear programming (ILP), respectively. The stopping criteria is set to achieve a solution with error less than 0.1% in each iteration. We calculate the Frobenius distance and the Chebyshev distance between **X** and **A** from IQP and ILP and compare with those given by the APM, the Newton method and the SDM, which are all designed for minimising the F-norm. The APM and the Newton method work very fast and can both achieve solutions with accuracy tolerance far lower than 0.0001, therefore the experiment settings have no big impact on testing results. We just use the default settings by the authors here (Higham, 2002; Qi and Sun, 2006). The results are shown in Tables 1 and 2 for the F-norm and the Chebyshev norm, respectively. Best performance in each group are in bold faces.

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	**0.3420**	**0.8341**	**5.0057**	**13.0331**	**20.6889**
APM	**0.3420**	**0.8341**	**5.0057**	**13.0331**	**20.6889**
SDM	0.3537	0.8602	5.1716	13.5887	21.5025
IQP	**0.3420**	0.8352	5.0245	13.2406	21.2224
ILP	0.3819	1.0815	7.6554	15.9193	24.9860

Table 1: The F-norm $||\mathbf{A}-\mathbf{X}||_{\mathbf{F}}$ obtained from different algorithms on different dimensions of the initial matrix $\mathbf{A}$.

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	0.1313	0.2128	0.6473	0.7605	0.8313
APM	0.1313	0.2128	0.6473	0.7605	0.8313
SDM	0.1416	0.2265	0.6225	0.7278	0.7512
IQP	0.1304	0.2131	0.6245	0.7314	0.8397
ILP	**0.0854**	**0.1375**	**0.4293**	**0.4310**	**0.4805**

Table 2: The Chebyshev norm $||\mathbf{A}-\mathbf{X}||_{\mathbf{C}}$ obtained from different algorithms on different dimensions of the initial matrix $\mathbf{A}$.

To give a more clear view on how these algorithms compare with each other, we use the Frobenius distance and the Chebyshev distance obtained from the Newton method as a benchmark and calculate the relative change of the two distances obtained from other algorithms in percentage. For example, the relative change of the Frobenius distance obtained from the IQP algorithm compared with the Newton method is given by $\frac{||\mathbf{A}-\mathbf{X}_{\text{IQP}}||_{\mathbf{F}}-||\mathbf{A}-\mathbf{X}_{\text{Newton}}||_{\mathbf{F}}}{||\mathbf{A}-\mathbf{X}_{\text{Newton}}||_{\mathbf{F}}} \times 100\%$. In Tables 3 and 4, a negative percentage means that the optimal solution obtained from this method is better than from the Newton method in the corresponding norm while a positive percentage represents a worse result.

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	**0**	**0**	**0**	**0**	**0**
APM	**0**	**0**	**0**	**0**	**0**
SDM	3.43	3.13	3.31	4.26	3.93
IQP	0.02	0.14	0.3763	1.59	2.58
ILP	11.67	29.66	52.93	22.15	20.77

Table 3: Relative change in the F-norm $||\mathbf{A}-\mathbf{X}||_{\mathbf{F}}$ compared with the Newton method in percentage.

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	0	0	0	0	0
APM	0	0	0	0	0
SDM	7.79	6.45	-3.84	-4.30	-9.64
IQP	-0.70	0.16	-3.52	-3.82	1.01
ILP	**-34.98**	**-35.37**	**-33.68**	**-43.33**	**-42.21**

Table 4: Relative change in the Chebyshev norm $||\mathbf{A}-\mathbf{X}||_{\mathbf{C}}$ compared with the Newton method in percentage.

Furthermore, we report the computing time of each run in Table 5. Lowest running times are in bold faces.

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	0.0024	0.0038	0.0058	0.0062	0.0350
APM	0.0005	0.0010	0.0068	0.0281	0.0615
SDM	**0.0001**	**0.0001**	**0.0003**	**0.0008**	**0.0025**
IQP	0.0315	0.1260	3.84	546	8180
ILP	0.0206	0.0937	2.35	719	5661

Table 5: Running time of different algorithms on different dimensions of the initial matrix $\mathbf{A}$ (in seconds).

It can be easily concluded from Table 3 (and also see Table 5 for efficiency) that both APM and the Newton method give solutions with the minimum Frobenius distance, which they aim to minimise, while the Newton method works slightly faster in high dimensions. (Further experiments show that the ALD approach also gives the same optimal solutions, which we omit in the above tables.) The SDM on the other hand is not as accurate as the Newton method and APM but is the fastest algorithm overall as it does not require iterations. The IQP method gives solutions slightly worse than the Newton method in terms of the F-norm but the difference is not significant, especially in low dimensions. These results from IQP can be improved by setting a lower tolerance and hence increasing the number of iterations but will cost more time to converge. The ILP method is designed to minimise with respect to the Chebyshev norm and therefore, does not give solid results in the F-norm as other algorithms mentioned above. However, the nearest correlation matrix generated by the ILP method has a much lower Chebyshev distance than by the Newton method which intends to minimise the F-norm (see Table 4). This result indicates that minimising over the Chebyshev norm can indeed be a good alternative for PSDisation since considering the F-norm will potentially result in huge deviations in some entries of $\mathbf{X}$, which are not favoured by some insurance companies (Cutajar et al., 2017). It is worth to mention that both IQP and ILP methods take a long time to run when the dimension of $\mathbf{A}$ is high, this could potentially be improved by optimising the QP and LP solvers in the future work.

To illustrate the flexibility of our IQP algorithm and its potential to achieve a good accuracy, our next step is to test when weights are assigned to each coefficients, *i.e.* to work under the H-norm. Similar as before, we generate initial matrix $\mathbf{A}$ of different dimensions. This time, we apply the ALD method and our IQP algorithm to find the nearest correlation matrix to $\mathbf{A}$ under the F-norm and the H-norm, respectively, where each entry of the weight matrix $\mathbf{H}$ is randomly generated from uniform distribution. We also test these methods on PSDisation problems with constraints, that is, we let $x_{ij} = 0$ if $|a_{ij}| < 0.1$, we also request $x_{ij} > 0$ if $a_{ij} > 0.5$ and $x_{ij} < 0$ if $a_{ij} < -0.5$. Table 6 shows the optimal values under different settings.

Dimension n				10	20	40	70
# of equality constraints (=0)				4	19	74	263
# of inequality constraints (<0/>0)				25	92	367	1208
Optimal solution	Unconstrained	F-norm	ALD	**2.5032**	**6.2625**	**14.8958**	**30.2514**
			IQP	2.5060	6.2789	15.1565	30.7230
		H-norm	ALD	**1.6149**	**3.9136**	**10.0493**	**20.3044**
			IQP	1.6188	3.9360	10.2596	20.8868
	Constrained	F-norm	ALD	**2.5401**	**6.4399**	**15.2334**	**30.8318**
			IQP	2.5411	6.4562	15.3771	31.2218
		H-norm	ALD	1.6631	4.0789	10.4666	**20.9562**
			IQP	**1.6501**	**4.0707**	**10.4355**	21.1677

Table 6: Optimal solution by ALD and IQP for constrained and unconstrained PSDisation problems under F-norm and H-norm on different dimensions of the initial matrix **A**.

It can be concluded that both ALD and IQP work effectively on constrained/unconstrained PSDisation problems based on F-norm or H-norm. Our IQP algorithm runs slowly when dimension is high and produces slightly worse results than ALD, but the optimal values are still competitive. Our iterative algorithm shows more flexibility when the problem settings are combined with H-norm (IQP) and Chebyshev (ILQ) norm, where the ALD method becomes infeasible.

3.2. Experiments on Gradient Descent Method

In order to test the performance of the gradient descent method on PSDisation problems, we compare ALD, IQP, GD and SDM on unconstrained problems of different dimensions under F-norm. SDM is a trivial method that project the matrix onto the positive semidefinite cone directly, and therefore can be considered as a baseline for this experiment. Table 7 compares the converge times and optimal solutions of all four algorithms.

Dimension n		10	20	40	70	100	200
Converge time (unconstrained, F-norm)	ALD	0.0029s	0.0027s	0.0120s	0.0076s	0.0163s	0.0364s
	IQP	0.17s	1.52s	2min 19s	2h 29min	-	-
	GD	0.03s	0.06s	0.16s	0.07s	0.08s	0.26s
	SDM	**0.0003s**	**0.0004s**	**0.0007s**	**0.0011s**	**0.0017s**	**0.0047s**
Optimal solution (unconstrained, F-norm)	ALD	**2.5032**	**6.2625**	**14.8958**	**30.2514**	**44.9199**	**96.3681**
	IQP	2.5060	6.2789	15.1565	30.7230	-	-
	GD	2.5316	6.2754	15.0992	30.8536	46.0570	98.7712
	SDM	2.5667	6.5599	15.5756	31.3282	46.5758	99.3325

Table 7: Converge time and optimal solution by ALD, IQP, GD and SDM for unconstrained PSDisation problems under F-norm on different dimensions of the initial matrix **A**.

As shown in Table 7, GD provides comparable results with ALD. Though slightly slower than ALD, GD works much more efficient than general SDP solvers. Furthermore, we want to illustrate the advantages of gradient descent (GD) method over the ALD algorithm. Consider the situation where in practice, the correlation matrices obtained by some insurance companies are usually inaccurate. Therefore, we expect our PSDisation algorithm to work better when error exists in the input matrix.

In the following experiment, we randomly generate a correlation matrix $\mathbf{A}$ that is not positive semidefinite. Noise from normal distribution is then added to each non-diagonal entry of $\mathbf{A}$ so that $\mathbf{A}'$ represents the measured correlation matrix. We use GD, ALD and IQP to calculate the nearest positive semidefinite correlation matrix $\mathbf{X}$ to $\mathbf{A}'$ under the F-norm and compare $||\mathbf{A}-\mathbf{X}||_{\mathbf{F}}$, the distance between $\mathbf{X}$ and the actual initial matrix $\mathbf{A}$.

	$\|\|\mathbf{A}'-\mathbf{X}\|\|_{\mathbf{F}}$	$\|\|\mathbf{A}-\mathbf{X}\|\|_{\mathbf{F}}$
ALD	**4.43**	5.70
IQP	4.45	5.72
GD	4.60	**5.49**

Table 8: Average Frobenius distance from $\mathbf{X}$ to $\mathbf{A}$ and $\mathbf{A}'$ by different algorithms over 100 tests in experiment settings.

Table 8 shows the average results of the three algorithms tested on 100 different initial matrices of dimension 40 with noise of standard deviation 0.2. It can be concluded that while GD provides slightly inaccurate solution to the PSDisation optimisation problem, it is less influenced by noise, and thus lead to better results than ALD overall. In all 100 testing examples, GD gives matrix $\mathbf{X}$ that is closer to the initial matrix $\mathbf{A}$. In practice, GD can be more resilient to noise with large standard deviation than ALD. Details of the experiments are given in Figure 1.

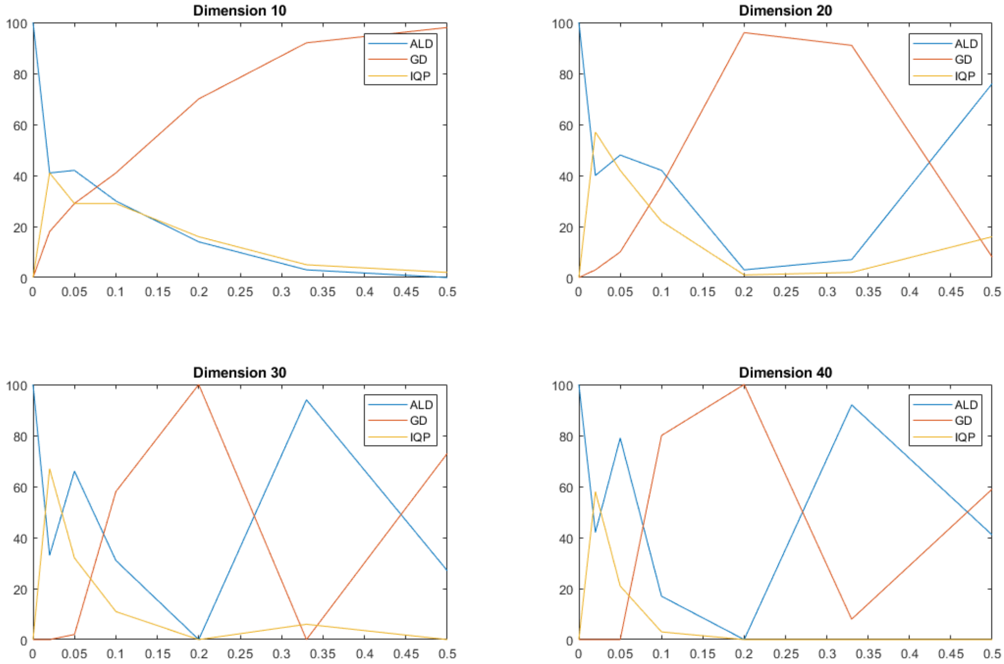

Figure 1: Percentage of best performance by ALD, GD and IQP versus standard deviation of the noise in PSDisation tests with different dimensions of the initial matrix $\mathbf{A}$.

Best performance is decided for giving the shortest distance between $\mathbf{X}$ and $\mathbf{A}$, where $\mathbf{X}$ is calculated by applying different PSDisation algorithms on noised matrix $\mathbf{A}'$. Tests are done for matrices of different dimensions and noise of different levels.

Despite the fact that ALD performs best on accurate initial matrices, IQP and GD can be good alternatives when the input comes with noise, especially for matrices with large dimension. IQP works well when the noise is small, *e.g.* in 67% of our tests IQP gives the smallest $||\mathbf{A} - \mathbf{X}||_{\mathbf{F}}$ for matrices with dimension 30 and noise with standard deviation 0.02. When the noise becomes slightly larger (for standard deviation between 0.1 and 0.2), GD performs significantly better.

3.3. Experiments on Simulated Data for a More Realistic Scenario

We present results of comparisons between ALD, IQP, GD and SDM on PSDisation problems in more realistic settings under F-norm. Without knowing the actual correlations, the way European Commission create PSD matrix for Solvency Capital Requirement insurance models is by picking the best possible choices for the correlation matrix entries from the set $\{-75\%, -50\%, -25\%, 0\%, 25\%, 50\%, 75\%\}$. The correlation matrix $\mathbf{A}$ estimated this way is usually not PSD, but certain entries of $\mathbf{A}$ can be expected to be positive or negative as actuaries expect some risks to be positive or negative correlated based on domain knowledge.

Inspired by the above ideas, our experiment is designed as follows. Firstly a PSD matrix $\mathbf{A}_t$ is generated which we assume is the true correlation matrix. Then 250 observations are generated from t-distribution with 3 degrees of freedom and are used to calculate sample correlation matrix $\mathbf{A}_s$. We obtain our estimated correlation matrix $\mathbf{A}'_s$ by rounding each entry of the sample correlation matrix $\mathbf{A}_s$ to 25%. We also round the true correlation matrix $\mathbf{A}_t$ to 25% to get $\mathbf{A}'_t$. Now $\mathbf{A}'_s$ and $\mathbf{A}'_t$ are not PSD. We perform unconstrained PSDisation on $\mathbf{A}'_s$ under F-norm and the solution is denoted as $\mathbf{X}$. We also test constrained PSDisation by ALD and IQP. The constrains are set so that entries of $\mathbf{X}$ have same signs as $\mathbf{A}_s$, which we assume are available to actuaries as domain knowledge.

The tests are done repeatedly on matrices of dimension 10, 20, 30 and 40. The constrained PSDisation are done using ALD, IQP, GD and SDM. In our experiment settings, around 85% to 89% of the non-diagonal entries of $\mathbf{X}$ have different signs from $\mathbf{A}_s$. In the constrined settings, both ALD(con) and QP(con) can 100% satisfy the same sign constrains, either strictly or to some small tolerance. The results are shown in Tables 9 and 10.

	n	**10**	**20**	**30**	**40**
$\vert\vert\mathbf{A}_s' - \mathbf{X}\vert\vert_\mathbf{F}$	**ALD**	**0.1387**	**0.3940**	**0.6991**	**1.0320**
	IQP	0.1388	0.3945	0.7003	1.0347
	GD	0.1453	0.4198	0.7512	1.1252
	SDM	0.1462	0.4247	0.7633	1.1479
	ALD(con)	0.1421	0.4081	0.7262	1.0735
	IQP(con)	0.1421	0.4086	0.7276	1.0760
$\vert\vert\mathbf{A}_s - \mathbf{X}\vert\vert_\mathbf{F}$	**ALD**	0.6429	1.2817	1.8862	2.4612
	IQP	0.6429	1.2820	1.8870	2.4633
	GD	0.6433	1.2837	1.8910	2.4796
	SDM	0.6440	1.2887	1.9019	2.5017
	ALD(con)	0.6351	**1.2499**	**1.8217**	**2.3611**
	IQP(con)	**0.6350**	1.2501	1.8223	2.3624

Table 9: Average Frobenius distance from $\mathbf{X}$ to $\mathbf{A}_s'$ and $\mathbf{A}'{}_s$ by different algorithms over 500 tests in experiment settings.

	n	**10**	**20**	**30**	**40**
$\vert\vert\mathbf{A}_t' - \mathbf{X}\vert\vert_\mathbf{F}$	**ALD**	1.6960	3.5396	5.2255	7.0881
	IQP	1.6960	3.5398	5.2261	7.0896
	GD	1.6851	3.4839	5.1133	6.8849
	SDM	**1.6814**	**3.4640**	**5.0635**	**6.7970**
	ALD(con)	1.6923	3.5240	5.1953	7.0412
	IQP(con)	1.6923	3.5240	5.1959	7.0425
$\vert\vert\mathbf{A}_t - \mathbf{X}\vert\vert_\mathbf{F}$	**ALD**	1.5294	3.2316	4.7367	6.4601
	IQP	1.5294	3.2318	4.7373	6.4617
	GD	1.5169	3.1710	4.6136	6.2364
	SDM	**1.5128**	**3.1499**	**4.5599**	**6.1427**
	ALD(con)	1.5252	3.2135	4.7015	6.4052
	IQP(con)	1.5252	3.2138	4.7022	6.4067

Table 10: Average Frobenius distance from $\mathbf{X}$ to $\mathbf{A}_t'$ and $\mathbf{A}'{}_t$ by different algorithms over 500 tests in experiment settings.

It can be easily conclude from Table 9 that ALD is the best in minimising the objective, $||\mathbf{A}_s' - \mathbf{X}||_\mathbf{F}$, while IQP could produce very similar results. On the other hand, GD gives slightly worse solutions than ALD and IQP, but still better than baseline method SDM. Comparing $||\mathbf{A}_s - \mathbf{X}||_\mathbf{F}$, unsurprisingly, we see that constrained PSDisation show its advantages if prior information is available. While ALD allows restrictions on upper or lower bounds of entries of $\mathbf{X}$, IQP could potentially allow any linear constrains and thus can be more flexible. Non-linear constrains can also be set if efficiency is not the main focus. For example, we could restrict entries of $\mathbf{X}$ to be multiples of 25%. This leads to iterative mixed integer programmings, which is beyond the scope of this paper.

On the other hand, comparing $||\mathbf{A}_t' - \mathbf{X}||_\mathbf{F}$ and $||\mathbf{A}_t - \mathbf{X}||_\mathbf{F}$ in Table 10, SDM and GD constantly give PSD matrix $\mathbf{X}$ that are closer to the true correlation matrix $\mathbf{A}_t$. Therefore, we draw the conclusion that though relatively inaccurate in minimising the objective function, $||\mathbf{A}_s' - \mathbf{X}||_\mathbf{F}$, in practice, SDM and GD are more resilient to noise generated in real data, which coincides with the our observations in the previous part.

3.4. Experiments for Real Financial Data

We now present an example of PSDisation applications with real financial data. We randomly select 50 new public companies from NASDAQ stock market and calculate the pairwised correlation of the daily asset return over the last thirteen years. The result correlation matrix $\mathbf{A}$ is non-PSD with negative eigenvalues due to missing values for each stock at certain periods of time. We perform PSDisation on calculated correlation matrix $\mathbf{A}$ to get $\mathbf{X}$ and find the Frobenius distance and the Chebyshev distance between $\mathbf{A}$ and $\mathbf{X}$. The same experiment is done on mixed choices of 25 NASDAQ stocks and 25 SP500 companies, where we tested two sets of mixtures (random selections). The results are shown in Tables 11, 12 and 13.

	$\|\|\mathbf{A}-\mathbf{X}\|\|_\mathbf{F}$	$\|\|\mathbf{A}-\mathbf{X}\|\|_\mathbf{C}$
IQP	0.3765	0.0872
ALD	**0.3764**	0.0886
GD	0.3958	**0.0836**
SDM	0.4085	0.0931

Table 11: Frobenius distance and Chebyshev distance between $\mathbf{A}$ and $\mathbf{X}$ by different algorithms on NASDAQ stocks return correlation matrix.

	$\|\|\mathbf{A}-\mathbf{X}\|\|_\mathbf{F}$	$\|\|\mathbf{A}-\mathbf{X}\|\|_\mathbf{C}$
IQP	0.2269	**0.0542**
ALD	**0.2268**	0.0547
GD	0.3958	0.0602
SDM	0.4085	0.0620

Table 12: Frobenius distance and Chebyshev distance between $\mathbf{A}$ and $\mathbf{X}$ by different algorithms on a mixture of NASDAQ and SP500 stocks return correlation matrix (set 1).

	$\|\|\mathbf{A}-\mathbf{X}\|\|_\mathbf{F}$	$\|\|\mathbf{A}-\mathbf{X}\|\|_\mathbf{C}$
IQP	0.3427	**0.0647**
ALD	**0.2542**	0.0651
GD	0.3872	0.0821
SDM	0.4018	0.0848

Table 13: Frobenius distance and Chebyshev distance between $\mathbf{A}$ and $\mathbf{X}$ by different algorithms on a mixture of NASDAQ and SP500 stocks return correlation matrix (set 2).

It can be concluded from Tables 11 and 12 that the optimal minimum Frobenius distance solution by our IQP method is close to the state-of-art ALD method, while maintaining a lower Chebyshev distance. This ensures that the maximum deviation of correlation for each pair of stocks by IQP is smaller than by ALD. We also noticed that in this real financial data experiment, the speed of convergence of IQP is faster and the optimal solution is closer to ALD comparing with those in previous simulated data. However, according to Table 13, it is worth to mention that the performance of our IQP can sometimes be not as expected for some data, of which the reason remains to be determined. In addition, GD performs slightly better than the baseline method SDM in this stocks data experiment but is still incomparable with ALD or IQP.

3.5. Examples on Applications in PCA

Principal component analysis (PCA) (Pearson, 1901) is a popular statistical tool to reduce the dimension of the dataset while maintaining as much information as possible from the data, which helps visualising the data or performing further actions in different fields of studies. The covariance matrix of the features is calculated before it turns into an eigenvalue problem (Jolliffe, 2002). A standardisation of the data is usually needed when the features are in different scales or, alternatively, the correlation matrix can be used instead of covariance matrix. If missing values present in the dataset, a standard way is to delete all instances containing missing values and then calculate the correlation matrix. However, ignoring such instances means that less information can be used from the whole dataset, which may lead to a decrease of the quality of PCA. We seek to use as much information as possible from the dataset by calculating the correlation between attributes pairwisely. This may cause the resulting correlation matrix to be non-PSD and therefore a PSDisation process is needed.

We use the *wine* data available from the UCI Machine Learning Repository (Dua and Graff, 2017) as an example. It consists of 13 attributes of chemical analysis of wines from different cultivars in Italy. The features are in different scales so it makes sense to use the correlation matrix in PCA. We first drop 20% of the values from the dataset randomly so that it has missing values. We calculate the pairwised correlation matrix which is non-PSD and thus invalid. Then PSDisation is performed on the invalid correlation matrix, after which eigenvectors are calculated to transfer the standardised data with missing values. The variance of each principal component of the transformed data is reported in Table 14. For this dataset all PSDisation methods give the same results, so we just choose our IQP algorithm for PSDisation in this example. We also perform PCA according to the correlation matrix calculated by omitting all instances that have missing values. Figure 2 shows the cumulative sum of the variance for each principal component of the transformed data by different methods, where *all PCA* stands for PCA done with all instances with missing values omitted and *pairwise PCA* stands for PCA using pairwised correlation matrix with PSDisation.

Principal Component	**all PCA**	**pairwise PCA**
1	3.1258	**3.2692**
2	1.5018	**1.6485**
3	0.7359	**1.0723**
4	0.6901	**0.7908**
5	**0.7454**	0.7164
6	0.5511	**0.5894**
7	**0.7294**	0.5543
8	**0.4103**	0.3897
9	**0.4756**	0.3526
10	**0.3532**	0.2875
11	**0.3712**	0.2895
12	**0.3508**	0.2604
13	**0.3436**	0.1637

Table 14: Variance of each principal component in the transformed wine data.

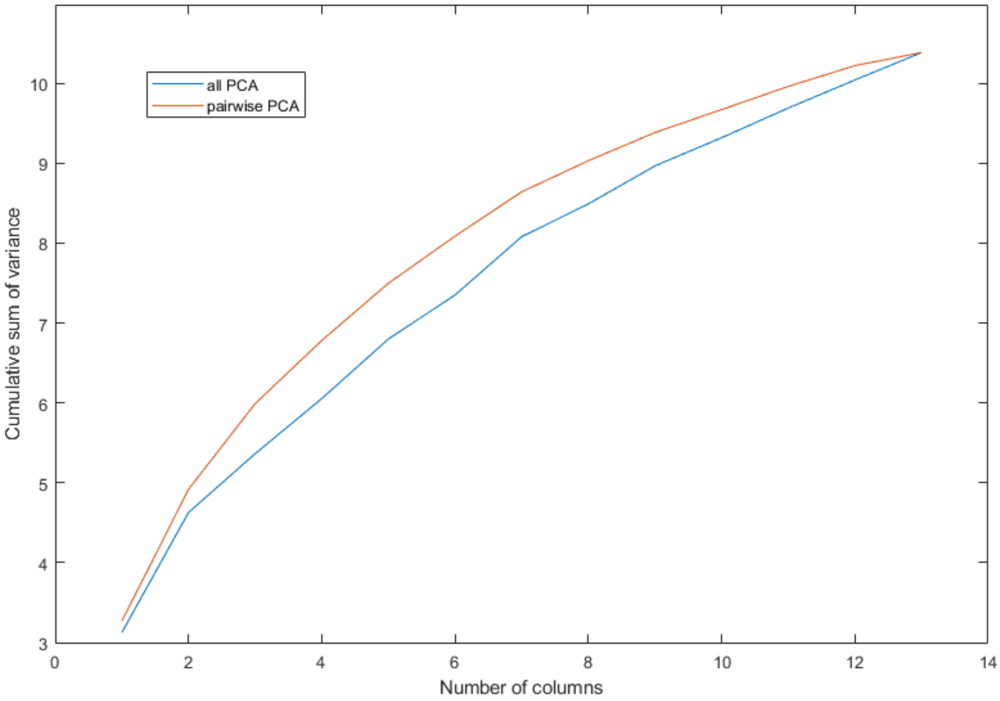

Figure 2: Cumulative sum of the variance of each principal component in the transformed wine data.

Table 14 indicates that with the use of pairwised correlation, the transformed data has higher variance in the first several principle components than all PCA. Therefore, more important information can be captured by the first few principal components via pairwise PCA.

Features	**PCA**	**pairwise PCA**	**all PCA**
Flavanoids	**0.4229**	**0.4212**	**0.3793**
Total phenols	**0.3947**	**0.3753**	**0.3494**
OD280/OD315 of diluted wines	**0.3762**	**0.3633**	0.3334
Proanthocyanins	0.3134	0.3046	0.1521
Nonflavanoid phenols	0.2985	0.2872	0.2453
Hue	0.2967	0.3090	0.2823
Proline	0.2868	0.3022	**0.3558**
Malic acid	0.2452	0.2237	0.2489
Alcalinity of ash	0.2393	0.2685	0.3481
Alcohol	0.1443	0.1886	0.3408
Magnesium	0.1420	0.1640	0.1399
Color intensity	0.0886	0.0550	0.0809
Ash	0.0021	0.0010	0.0893

Features	**PCA**	**pairwise PCA**	**all PCA**
Color intensity	**0.5300**	**0.5522**	**0.5696**
Alcohol	**0.4837**	**0.4429**	0.2506
Proline	**0.3649**	**0.3603**	0.2577
Ash	0.3161	0.3533	**0.3538**
Magnesium	0.2996	0.2679	**0.3946**
Hue	0.2792	0.2824	0.3146
Malic acid	0.2249	0.2240	0.0492
OD280/OD315 of diluted wines	0.1645	0.1697	0.2766
Total phenols	0.0650	0.0788	0.0717
Proanthocyanins	0.0393	0.0280	0.2065
Nonflavanoid phenols	0.0288	0.0769	0.0902
Alcalinity of ash	0.0106	0.0287	0.1363
Flavanoids	0.0034	0.0049	0.1147

Table 15: Weight of each feature assigned by the 1_{st} and 2_{nd} principal components in the wine data by PCA, pairwise PCA and all PCA. Top: Weights of features in first principal component. Bottom: Weights of features in second principal component.

Furthermore, Table 15 summaries the weight of each feature that the first and second principle components contain by PCA on full data and pairwise PCA and all PCA on data with missing values. The weights of the first three features with highest weights are marked in bold. It can be seen that PCA and pairwise PCA assign similar weights to each feature in their first and second priciple components. In contrast, while all PCA selects the same feature with highest weights as original PCA, the weights of other features it gives are generally much further away than pairwise PCA. We may conclude that pairwise PCA preserves as much information as possible from data with missing values and thus produces more similar results as PCA from the full dataset than all PCA.

4. Conclusion

In this paper, we discussed the PSDisation problem for actuarial analysis in order to find the nearest correlation matrix. We first stated the general formulation of the PSDisation problem before several state-of-the-art methods were briefly summarised. We then proposed two new approaches to tackle the problem, that is, the iterative quadratic/linear programming method and the gradient descent method. IQP/ILP is a flexible algorithm that approximate the nearest correlation matrix by solving a series of optimisations with linear constraints. The accuracy and flexibility of the algorithm were shown by experiments on randomly generated initial matrices, stocks daily return correlation matrix and data for PCA purpose. We concluded that despite more time-consuming than traditional methods, the IQP method could achieve an optimal solution similar to the APM, Newton or ALD method when working with the F-norm. What's more, it is crucial to understand that our formulation of the problem is very flexible and thus can handle norms of any choice. In our experiments, the ILP and IQP method produce good results under the Chebyshev norm and the H-norm, respectively. It is also possible to combine different norms in our formulation (*e.g.* to minimise $\alpha||\mathbf{A}-\mathbf{X}||_{\mathbf{F}}^2 + (1-\alpha)||\mathbf{A}-\mathbf{X}||_{\mathbf{max}}^2$ where α is a weighting parameter). Future works can be done to increase the speed of solving the quadratic or linear optimisation problem in each iteration to increase the efficiency in order to cope with larger problems. The GD method repeats the followings, taking a step in the opposite direction of the gradient, projecting the matrix onto the positive semidefinite cone via spectral decomposition and scaling the matrix so that the diagonal is 1. This method is effective in unconstrained PSDisation problems with respect to F-norm. Experiments in Section 3 shows the efficiency and robustness of the algorithm. Future attempts to modify the GD algorithm can be considered so that it may work with different norms or solve problems with more constraints.

References

Ahmadi, A. A. and Hall, G. (2017). Sum of squares basis pursuit with linear and second order cone programming. *Algebraic and Geometric Methods in Discrete Mathematics*, 685:27–53.

Cantrell, C. D. (2000). *Modern mathematical methods for physicists and engineers.* Cambridge University Press.

Cutajar, S., Smigoc, H., and OâHagan, A. (2017). Actuarial risk matrices: the nearest positive semidefinite matrix problem. *North American Actuarial Journal*, 21(4):552–564.

Deutsch, F. and Hundal, H. (1997). The rate of convergence for the method of alternating projections, ii. *Journal of Mathematical Analysis and Applications*, 205(2):381–405.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Higham, N. J. (2002). Computing the nearest correlation matrixâa problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343.

Higham, N. J., Strabic, N., and Sego, V. (2016). Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block. *SIAM Review*, 58(2):245–263.

Jolliffe, I. T. (2002). *Principal component analysis for special types of data*. Springer.

Milhaud, X., Poncelet, V., and Saillard, C. (2018). Operational choices for risk aggregation in insurance: Psdization and scr sensitivity. *Risks*, 6(2):36.

Pearson, K. (1895). Notes on regression and inheritance in the case of two parents. In *Proceedings of the Royal Society of London*, volume 58, pages 240–242.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

Qi, H. and Sun, D. (2006). A quadratically convergent newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 28(2):360–385.

Qi, H. and Sun, D. (2011). An augmented lagrangian dual approach for the h-weighted nearest correlation matrix problem. *IMA Journal of Numerical Analysis*, 31(2):491–511.

Rebonato, R. and Jäckel, P. (2011). The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. *Available at SSRN 1969689*.

Van Loan, C. F. and Golub, G. H. (1983). *Matrix computations*. Johns Hopkins University Press Baltimore.