Data-adaptive Principal Component Analysis for High Dimensional Data

Lingyu He

A thesis submitted for the degree of Doctor of Philosophy

in Statistics

The Australian National University

November 2020

Except where otherwise indicated, this thesis is my own original work.

Lingyu He

12 November 2020

To my beloved family.

Acknowledgments

I would like to thank Research School of Finance, Actuarial Studies and Statistics (RSFAS) at the Australian National University (ANU) and the China Scholarship Council for providing me an opportunity and scholarships to do my research and meet with prestigious researchers. During the period of PhD studies, I have been helped by many individuals. I would like to express my gratitude to those who made this thesis possible.

Firstly, I would like to take this opportunity to express my sincere appreciation to my supervisors Dr. Yanrong Yang, Professor Steven Roberts, and Professor Eric Stone for their encouragement, support, and advice throughout this period. In particular, I want to thank Dr. Yanrong Yang for her invaluable guidance in all stages of my PhD studies. Dr. Yanrong Yang has not only helped me solving research problems but also encouraged me in developing novel research ideas. She has always been very generous and patient in supervising me to pursue the best. I sincerely thank Professor Steven Roberts and Professor Eric Stone for providing their expertise in fields of statistics. Their wide knowledge and expert thinking have inspired me throughout my PhD studies.

It has been a great pleasure for me to work with the colleagues and fellows at RSFAS. In particular, I would like to thank Dr. Fei Huang and Mr. Jianjie with whom I worked closely for the goal of publishing Chapter 3 of this thesis. I appreciate the help of Associate Professor Timothy Higgins for his considerable help and guidance in PhD-related issues. I would also like to thank Dr. Tao Zou for the generous advices on researches. Moreover, thanks to my dear friends, Dr. Jiali Wang and Ms. Yuan Gao, for the great companion and support during my study.

Last but not the least, I wish to acknowledge the support and great love of my family, my husband, Daning; my mother, Yunxia; and my father Yuanjiang. They kept me going on and this work would not have been possible without their input.

Thank you all who helped me during my PhD journey.

Abstract

Among many well-designed techniques for dimension reduction, the Principal Component Analysis (PCA) is one of the most popular and applicable methods. In this thesis, we address the challenges encountered when modelling and forecasting the high-dimensional data with PCA related methods in three problems.

In Chapter 2, we propose a two-style factor model to improve the forecasting of high-dimensional time series. The model pursues two types of low-dimensional features for the original high-dimensional time series, with one type summarizes the common time-serial trend, and the other one represents the common variations. The two types of features benefit the forecasting and the model fitting, respectively. The dynamic PCA and the static PCA are utilized in a sequential way to estimate the features in the model. We show the proposed method enjoys good statistical performance and illustrate the advantages of it with various simulating studies. By modelling and forecasting the US mortality data, we show that our method provides more accurate forecasts, especially comparing to the Lee-Carter model, which is the most popular model in mortality analysis.

In Chapter 3, we continue to study the mortality modelling. Classical mortality models usually assume the factor loadings, which capture the relationship between age variables and latent common factors, are time-invariant. This assumption, however, is too restrictive in reality, as mortality datasets typically span a long period of time. In order to reflect the changing relationship between age variables and latent common factors, we introduce a factor model with time-varying factor loadings to model the mortality data. Accordingly, two forecasting methods are proposed for which the estimated time-varying factor loadings are predicted using local linear regression and inheriting historical value (the naive method), respectively. In the empirical data analysis and the simulation studies,

the proposed method can recover the time-varying factor loadings and significantly improve the mortality forecasting. As further study, we propose a method to estimate the optimal "boundary" between the short-term and long-term forecasting, which is favored by the two forecasting methods, respectively. In view of this, a hybrid forecasting method can be utilized, which consists of the local regression method before the optimal boundary and the naive method thereafter.

In Chapter 4, we propose a novel robust PCA for high-dimensional data in the presence of various kinds of heterogeneities, such as outliers, heteroscedastic noise, and heavy-tailed variables. The method is based on a characteristic-function-type of transformation. Besides the typical outliers, the proposed method has the unique advantage of dealing with heavy-tail-distributed data, whose covariances could be nonexistent (positively infinite, for instance). We show the merit and the cost of the method by studying the estimation accuracy of the reconstruction error and the impact of the transformation on a spiked covariance structure. In addition, simulation studies show the advantage of our method on data with heterogeneities. At last, we apply the method to classify mice with different genotypes in a biological study based on their protein expression data and find that our method is more accurate on identifying abnormal mice comparing to the standard PCA.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Principal component analysis (PCA) was firstly proposed and discussed by Pearson [1901] and Hotelling [1933] as a statistical dimension reduction method, which aimed to represent multivariate data onto a lower dimensional space, while minimizing the loss of information. Since developed, applications of PCA have been widely investigated in many areas such as physics, biology and economics, as discussed in Jolliffe [2002]. Thanks to the development of computer science in recent years, a vast number of data are being collected for statistical analysis and the dimension of data or number of variables are growing dramatically, see for example, Huang et al. [2010]; Hyndman et al. [2013]; Ando and Bai [2017]. Among many well-designed techniques for dimension reduction, PCA is one of the most applicable methods and it can be easily interpreted. Hence, as a powerful dimension reduction technique, PCA has once again became attractive to many researchers.

Nonetheless, the growth of the dimension has also lead to the increased complexity of the data collected (Fan et al. [2018a]). While PCA is still a powerful tool for reducing the dimension of such data, it faces challenges as the complexity of the data increases. To address these challenges, we adapt traditional PCA methods for high-dimensional data with different types and structures in this thesis. More specifically, the following statistical challenges related to PCA are considered:

- Chapter 2: extracting low-dimensional features which are ideal for the forecasting of mortality data;
- Chapter 3: recovering the changing relationship between age variables and common features in age-specific mortality data to improve model fitting and forecasting;
- Chapter 4: reducing dimension when high dimensional data contain varies kinds of heterogeneities, such as outliers and heavy-tail-distributed variables.

The outline of the thesis is as follows. Note that each chapter uses their own mathematical notations.

In the rest sections of this chapter, we discuss the relationship between PCA and the factor model, provide literature on modelling and forecasting the age-specific mortality data using factor models, and review literature about dynamic PCA and robust PCA. We also point out our contributions in those sections.

In Chapter 2, we propose a two-style factor model to seek linear features that attain optimal forecasting of the US age-specific mortality data. The age-specific mortality data is a representative high-dimensional time series data, as it contains death rate for a given age at a specific year and the number of years is comparable to the number of age groups. The model pursues two types of low-dimensional features for the mortality data, with one type summarizes the common time-serial trend, and the other one represents the common variations. The dynamic PCA and the static PCA are utilized in a sequential way to estimate the features in this model, which allows the two types of features benefit the forecasting and the model fitting, respectively. We show the proposed method enjoys good statistical performance in the sense that both types of features have equally fast convergence rates. Various simulating studies illustrate the advantages of the new method over the standard PCA and the dynamic PCA. We use rolling-window method to evaluate the forecasting performance of our method comparing to

other classical methods on the US mortality data. It shows that the two-style factor model do provides better forecasts on the US mortality data, especially comparing to the Lee-Carter model (Lee and Carter [1992]), which is the most popular model in mortality analysis.

In Chapter 3, we continue to study the mortality modelling. Many mortality models, such as the Lee-Carter and its variants, assumed the relationship between a given age and mortality level (the common factor in factor model) is time-invariant. This assumption is usually too restrictive in reality as mortality datasets typically span a long period of time. Driving forces such as medical improvement of certain diseases, environmental changes, and technological progress may influence the relationship of different variables significantly. To recover the changing relationship between age variables and latent common factors, we introduce a factor model with time-varying factor loadings to model the mortality data. The time-varying factor loadings are estimated by a local version of PCA proposed by Su and Wang [2017], which is a semi-parametric kernel method. Based on the time-varying factor model, two forecasting methods are proposed to extrapolate the time-varying factor loadings into feature. One method uses local linear regression and the other inherits the most recent historical value (the naive method). In the simulation studies, we show that the time-varying model provides more accurate forecasts than the classical model when the true factor loading is changing over time. This is because the time-varying model is able to recover the changing factor loading while the classical model can not. We show the proposed method can significantly improve the mortality forecasting by comparing it to other mortality models with mortality data of varies countries and forecasting horizons. As further study, we propose a novel approach based on change point analysis to estimate the optimal "boundary" between the short-term and long-term forecasting, which is favored by the two forecasting methods, respectively. In view of this, a hybrid forecasting method can be utilized, which consists of the local regression method before the optimal boundary and the naive

method thereafter.

In Chapter 4, we propose a novel robust PCA for the dimension reduction of high-dimensional data in the presence of various kinds of heterogeneities, such as outliers, heteroscedastic noise, and heavy-tailed variables. The estimation of the standard PCA is based on covariance matrix, while the sample covariance matrix is sensitive to outliers. Devlin et al. [1981] showed that the usual estimator of the covariance matrix can lead to misleading estimation of the principal components under the presence of outlying observations. Furthermore, heavy-tail-distributed data also poses challenges to the inference of standard PCA, as their covariance matrix could be nonexistent (positively infinite, for example). As a result, the standard PCA is not robust to the presence of outliers or heavy-tailed variables. To address the above difficulties simultaneously, in Chapter 4, we propose a novel robust PCA based on a transformation related to characteristic function. The transformed sample covariance matrix shrinks the impact of outliers or heavy-tailed errors. We show the method is more robust than the classical PCA, in the sense that it has small excess risk of the reconstruction error even applied to extremely heavy-tailed data whose covariances could be nonexistent. We also study the behavior of the large eigenvalues under a spiked covariance model to illustrate the impact of the transformation on the spike structure. In addition, we show the advantages of our method in the sense of the mean squared reconstruction error compared with the standard PCA by a variety of simulations. At last, we apply the method to a mice data from Higuera et al. [2015]. The data consists protein expressions of mice with different genotypes in a biological study. The robust PCA provides more accurate classifications for normal and abnormal mice comparing to the standard PCA.

Appendix A provides additional simulations and the proof for Chapter 2. Appendix B provides additional empirical applications for Chapter 3.

Next, let us review related methods and literature in the following sections.

1.2 Factor model and PCA for high dimensional data

Factor analysis and principal component analysis (PCA) are two widely used statistical methods. It is shown in Fan et al. [2013] that PCA can be used for the factor analysis in the presence of spiked eigenvalues under high-dimensional setting. In Chapter 2 and 3, we use factor models to model the underlying high-dimensional data and adapt PCA methods to estimate corresponding factors and factor loadings. In this section, we discuss some mathematical details of PCA and factor model, as well as their relationship. One can refer to Fan et al. [2013, 2018a] and Fan et al. [2018b] for more detailed discussion.

We first summary the standard PCA here. Let $\boldsymbol{y} = (y_1, \ldots, y_p)^\top$ be a random vector taking values in $\mathbb{R}^p$ with mean zero and covariance matrix $\boldsymbol{\Sigma}$. With this formalism, standard PCA seeks projection direction vectors, $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k \in \mathbb{R}^p$, such that

$$\boldsymbol{\beta}_1 \in \underset{\|\boldsymbol{\beta}\|_2=1}{\operatorname{argmax}} \boldsymbol{\beta}^\top \boldsymbol{\Sigma} \boldsymbol{\beta}, \quad \boldsymbol{\beta}_2 \in \underset{\|\boldsymbol{\beta}\|_2=1, \boldsymbol{\beta} \perp \boldsymbol{\beta}_1}{\operatorname{argmax}} \boldsymbol{\beta}^\top \boldsymbol{\Sigma} \boldsymbol{\beta}, \quad \boldsymbol{\beta}_3 \in \underset{\|\boldsymbol{\beta}\|_2=1, \boldsymbol{\beta} \perp \boldsymbol{\beta}_1, \boldsymbol{\beta}_2}{\operatorname{argmax}} \boldsymbol{\beta}^\top \boldsymbol{\Sigma} \boldsymbol{\beta}, \ldots.$$

Mathematically, the optimal solution for $\{\boldsymbol{\beta}_i\}_{i=1}^R$ are the R eigenvectors corresponding to the top R eigenvalues of $\boldsymbol{\Sigma}$. Given $\boldsymbol{B}_R := (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_R)$, the goal of dimension reduction is achieved by projecting the original high dimensional data onto the low dimensional subspace $S \subset \mathbb{R}^p$ of dimension R ($R < p$) spanned by columns of $\boldsymbol{B}_R$. Since $\boldsymbol{B}_R$ captures the most variation in the dataset, the projected data $\boldsymbol{B}_R^\top \boldsymbol{y}$ approximately preserves the geometric properties of the original data, which are amenable to downstream statistical analysis (Fan et al. [2018a]). In applications, the unknown population covariance matrix $\boldsymbol{\Sigma}$ is replaced by the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$ in order to estimate the projection direction vectors.

Factor model is a frequently used method in actuarial, economic and financial studies (for example, Stock and Watson [2002]; Cairns et al. [2006]; Ando and Bai

[2017]). It aims to capture dependence across multivariate variables by assuming several "common factors" (Anderson [1963]; Bai and Ng [2002]; Lam et al. [2011]; Fan et al. [2018b]). Those common factors summarize the dependence of the whole multivariate data and the number of common factors are usually much smaller than the number of variables. In recent decades, literature such as Bai and Ng [2002]; Fan et al. [2013] studied the factor model under high-dimensional settings. A typical factor model for $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ (n observations of the random vector $\boldsymbol{y}$ defined in the last paragraph) is constructed as follows:

$$y_{ij} = \boldsymbol{\mu}_j + \boldsymbol{b}_j^\top \boldsymbol{f}_i + \varepsilon_{ij}, \quad i = 1, \ldots, n, \ j = 1, \ldots, p, \tag{1.2.1}$$

where y_{ij} is the j^{th} response of the i^{th} observation $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ip})^\top$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^\top$ is the intercept, $\boldsymbol{b}_j$ $(R \times 1)$ is vector of factor loadings, $\boldsymbol{f}_i$ $(R \times 1)$ is the vector of R common factors, and $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{ip})^\top$ is the error term independent of $\boldsymbol{f}_i$. In model 1.2.1, only y_{ij}'s are observable.

Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p)^\top$ $(p \times R)$ be the factor loading matrix. Then based on the model 1.2.1, $\boldsymbol{\Sigma}$, the covariance matrix of $\boldsymbol{y}$, is given by

$$\boldsymbol{\Sigma} = \boldsymbol{B} cov(\boldsymbol{f}_i) \boldsymbol{B}^\top + \boldsymbol{\Sigma}_\varepsilon, \quad \boldsymbol{\Sigma}_\varepsilon = (\sigma_{\varepsilon,jr})_{1 \le j,r \le p} = cov(\varepsilon_i). \tag{1.2.2}$$

It is easy to obtain that the columns of $\boldsymbol{B}$ are the eigenvectors of the matrix $\boldsymbol{B} cov(\boldsymbol{f}_i) \boldsymbol{B}^\top$ corresponding to its non-zero eigenvalues. The literature on high-dimensional factor models (Bai and Ng [2002]) usually assume that the R non-zero eigenvalues (R is assumed to small compared to p and n, e.g. fixed) of $\boldsymbol{B} cov(\boldsymbol{f}_i) \boldsymbol{B}^\top$ diverge at rate $O(p)$ and all the eigenvalues of $\boldsymbol{\Sigma}_\varepsilon$ are bounded as $p \to \infty$. The assumption holds when there is a spiked eigenvalue structure in $\boldsymbol{\Sigma}$, that is a non-negligible fraction of eigenvalues larger than the rest (Fan et al. [2018a]). Hence, under a spiked covariance structure, columns of $\boldsymbol{B}$ are close to the eigenvectors corresponding to the top R eigenvalues of $\boldsymbol{\Sigma}$, which

means $\boldsymbol{B}$ is close to $\boldsymbol{B}_R$ obtained by the PCA. Fan et al. [2013] showed that the PCA and factor analysis are approximately the same if $\|\boldsymbol{\Sigma}_\varepsilon\|_2 = o(p)$. In view of this, PCA can be used to estimate the factor model when $n, p \to \infty$. Therefore, in Chapter 2 of this thesis, we utilize PCA methods to estimate the factors and factor loadings in the proposed two-style factor model and study statistical properties of the estimators by assuming a spiked covariance structure under high-dimensional setting.

1.3 PCA for high-dimensional time series

We review literature about PCA for high-dimensional time series in this section. In many areas such as economics, finance and medical studies, the data interested for analysis are often serial dependent. For example, in bio-medical research, time improvement of the patients' treatment can be studied via longitudinal data, as the empirical study in Martinussen and Scheike [2000]; in finance, Ando and Bai [2017] tried to identify the sources of the co-movement of international stock returns. The mortality data studied in Chapter 2 and 3 is a representative high-dimensional time series data, which is important for forecasting death rate or life expectancy (Lee and Carter [1992]).

Traditionally, the application of PCA relying on the assumption that the observations are independent. However, when applied to time series data, the standard PCA, or static PCA, fails to take into account the potentially valuable information carried by the past values of the time series under study. Because mathematically the static PCA is only based on the variance-covariance matrix but not the auto-covariance. To amend this drawback, dynamic PCA has been developed as a remedy for time series. An early work of dynamic PCA can be found in Brillinger [1975], where the author estimated PCs based on the Fourier transforms of the eigenvectors of spectral matrices. That is to say, dynamic PCA involves both the variance-covariance matrix and time-lagged auto-

covariance matrix when solving the subspace that contains the most variations of the original data. In other words, the variation of data not only refers to the cross-sectional complexity of data but also the temporal dependence. Depending on the targets, different varieties of dynamic PCA have also been developed recently. Hörmann et al. [2015] developed and studied a dynamic functional PCA for functional data analysis, which extended the method of Brillinger [1975] in a functional setup; Lam et al. [2011]; Lam and Yao [2012] studied factor modeling for high-dimensional time series based on dynamic PCA and provided inference for relevant estimators; Peña and Yohai [2016] generalized the dynamic PCA to adapt data with non-stationarity, nonlinearity, or outliers.

Our work in Chapter 2 is closely related to Lam and Yao [2012]. In their method, only auto-covariance matrices are involved in estimating of factors and factor loadings, as they assumed the original high-dimensional time series come from a low dimensional factor process. However, when the original data are generated from a more complicated factor structure which contains both time-serial dependent factors and temporally unrelated factors, both variance-covariance and auto-covariance matrices should be involved. The two-style factor model proposed in Chapter 2 aims to handle such kind of factor structure and we show the subspace spanned by eigenvectors that maximize the covariance does not represent the same subspace extracted from auto-covariance. Following a two-step estimation procedure, we extract two types of factors: one type of factors summarize most of the temporal dependence and are extracted via eigendecomposition of auto-covariance matrices; the other type of factors represent most of the variations after eliminating the temporally dependent factors and are captured by variance-covariance matrix. Lam and Yao [2012] also investigated a two-step procedure for their model. Their two steps are both based on auto-covariance matrices, hence provide two kinds of factors with the ones from the first step have stronger strength, or faster convergence rate, than those from the second step. The two kinds of factors in the method of Chapter 2, however, are both

strong factors in the sense that they are established based on auto-covariance and variance, respectively. More details are in Chapter 2 and the corresponding Appendixes.

1.4 Mortality data and mortality forecasting

We briefly discuss mortality data and methods for mortality modelling and forecasting in this section. A comprehensive review of the methods since 1980 can be found in Booth and Tickle [2008]. Millossovich et al. [2018] summarized more recent developments and provided the R package to implement those models.

The mortality datasets studied in this thesis come from the Human Mortality Database(HMD 91). HMD contains original calculations of death rates and life tables for the populations in 40 countries and areas, as well as the input data used in constructing those tables. In Chapter 2, the data obtained from HMD includes the annual age-sex-specific information of the number of exposures to risk, the number of deaths, and the central death rate, for ages from 0 to 110+ (age 100 and above) during the period from 1933 to 2016 for the US population. In Chapter 3, besides the the US mortality data, we also study the age-specific mortality rates of other countries including Australia, Canada, France, Italy, and Japan. The age-specific mortality data consists of annually observations on death rates of a population under each age, hence the age variables are usually as many as the yearly observations for each age. In view of this, age-specific mortality rates are high-dimensional time series.

Mortality forecasting is important for various areas, such as actuarial science, demography, and government policymaking. Because age-specific mortality data are usually high dimensional, many existing stochastic mortality models follow the framework of factor models. Lee-Carter model (Lee and Carter [1992]) is one of the most influential such kind of methods. In Lee-Carter model, one common factor is extracted and defined as Mortality Index, and the factor loadings capture

the relationship between the age variables and the mortality index. Then the forecasting of the mortality rates is obtained by forecasting the mortality index with classical univariate time series model. The method is easy to implement and interpret, hence it is popular in mortality forecasting.

Since there is only one factor in the Lee-Carter model, Booth et al. [2002], Renshaw and Haberman [2003], and Yang et al. [2010] extended the Lee-Carter framework to incorporate more common latent factors for mortality modeling in different countries. Hyndman and Ullah [2007] generalized the Lee-Carter method under the functional data setting and also allowed more common factors. To handle outliers in the mortality index, Li and Chan [2005] combined the Lee-Carter model with time series outlier analysis and proposed an outlier-adjusted model. Additionally, the Cairns-Blake-Dowd (CBD) model, a prominent variant of the Lee-Carter model and introduced by Cairns et al. [2006], includes a cohort effect term in the Lee-Carter model. More recently, Richman and Wuthrich [2019] utilized Neural Network to extend the Lee-Carter model to multiple populations. With the similar purpose, Shang and Haberman [2020] proposed methods to forecast multiple functional time series in a group structure with the functional model in Hyndman and Ullah [2007].

The work in this thesis contribute to two aspects of mortality modelling and forecasting. Firstly, Chapter 2 aims to improve the mortality forecasting by seeking the most suitable factors. The goal is achieved by the newly proposed two-style factor model and the corresponding two-step estimation approach. Secondly, Chapter 3 focuses more on improving the fitting of the factor loadings, which also leads to more accurate forecasting. Time-varying factor model is applied to mortality forecasting in Chapter 3, which was not considered in mortality forecasting literature to the best of our knowledge. The assumption of time-invariant factor loadings in Lee-Carter model and its variants is not realistic for mortality data spanning a long period of time, hence time-varying model provides better model fitting by relaxing this assumption.

1.5 Robust PCA methods

There have been a lot of robust variants of PCA proposed in the statistical literature for dealing with the non-robustness issue encountered by applying the standard PCA. One can refer to Chapter 10.4 in Jolliffe [2002] for a comprehensive review of the robust estimation of principal components in the early decades and refer to Maronna et al. [2019] for an introduction of the robust statistics. Our robust PCA in Chapter 4 is more robust than the classical PCA in the sense that it performs well even when the covariance matrix of the data is inexistent.

In the rest of this section, we review the robust PCA methods in the literature and summarize them into several classes following the survey in She et al. [2016]. The largest group of robust PCA methods is *the robust covariance matrix based method*. It is a simple and natural idea to replace the sample covariance matrix with a robust covariance matrix estimate in the standard PCA and then extract the eigenvectors from this estimate as the projection directions. In fact, every new robust covariance matrix estimator has a new robust PCA method associated with it (Jolliffe [2002]). Some earlier work of the robust covariance matrix estimation can be found in Campbell [1980], Devlin et al. [1981], and Davies [1987], in which they achieved the robustness by utilizing the Mahalanobis distance, M-estimator, or S-estimator. Croux and Haesbroeck [2000] studied some such kinds of robust covariance matrix estimators by examining the influence function and efficiency for the results of PCA. Recently, with the amount of data getting large, researchers are more focusing on studying the covariance matrix under high dimensional settings. For example, Chen et al. [2018] introduced a concept called "matrix depth" and studied the convergence rate for the "deepest" covariance matrix, which was the proposed robust covariance matrix estimator, under high dimensional regimes; Fan et al. [2019] proposed a method to robustly recover covariance matrix using the factor model for high dimensional data; Avella-Medina et al. [2018] presented robust matrix estimators aiming to relax the usual sub-

Gaussian assumption and fit a much richer class of distributions. See also Fan et al. [2017]; Minsker [2018] and the references therein. There are some limitations of the above-mentioned methods. Firstly, as some of the estimations involve iterative or re-sample procedure, the computational cost is high, especially when the sample and dimension are both large. Secondly, in order to plug in the robust covariance matrix estimators to estimate the principal components, the population covariance matrix needs to exist. For some heavy-tailed distributions, however, the covariance matrix can be positive infinite, i.e. the population covariance matrix of the data does not exist. Hence, performing the standard PCA might be initially unreasonable. On the other hand, the proposed method in Chapter 4 solves this issue and is valid under the aforementioned situation.

Another important class is *the projection based method*. The initial idea of the standard PCA is to find directions which maximize the variance of the projected data, sequentially. Hence, another natural idea is replacing the scale measure of the projected data with a robust one. The projection-pursuit method proposed by Li and Chen [1985] utilizes a projection-pursuit index instead of the variance to measure the dispersion of the projections. While the original algorithm is computer intensive, several alternatives have been provided to improve the efficiency as well as the computation time of the projection based method, which include Rousseeuw and Croux [1993]; Croux and Ruiz-Gazen [1996, 2005], etc. In addition, Hubert et al. [2005] proposed a method called "ROBPCA" for high dimensional data, which combines the projection pursuit idea with the robust covariance matrix estimation.

More recently *the low rank matrix approximation based method* is popular in computer science. Triggered by the need of methods to deal with huge datasets in areas such as image processing and Web analysis, researchers in computer science have done a lot of work on PCA. They view the PCA as a low-rank matrix recovering problem and to achieve the robustness, robust loss functions are utilized in the optimization problem. For example, Wright et al. [2009] showed

that combining the solutions to nuclear norm minimization for low-rank recovery and l^1-minimization for error correction perfectly recovers a low-rank matrix from large but sparse errors with high probability. Candès et al. [2011] proved that under some suitable assumptions, it is possible to recover both the low-rank and the sparse components exactly by solving a convex program called Principal Component Pursuit. Further variations such as Zhao et al. [2014] considered more complicated noise component. For a comprehensive review of the robust subspace learning, one can refer to Bouwmans et al. [2017].

Our method proposed in Chapter 4 is different from the above methods. We project the original data onto another space to achieve the robustness. This kind of idea was also implemented by Locantore et al. [1999], in which they proposed robust method by projecting the original data onto the unit sphere (centered at the spatial median). Their method is specifically for a kind of functional image data, while our method is more general.

Chapter 2

Data-adaptive Dimension Reduction for Mortality Forecasting

2.1 Introduction

The age-specific human mortality data consists of observations on either the death numbers or the death rates of a population under each age, measured for each historical year. Accurate forecasting of mortality data plays a crucial role in demography and actuarial science. For instance, the life expectancy and present value of life annuity are highly related to the future mortality rates. According to the life table published by Social Security Administration [2019], from 2016 to 2095, the life expectancy, which is the average remaining years of life at a specific year, for a male aged 66 in the US will rise from 17.2 to 21.7 years. Meanwhile, the present value (price) of the corresponding life annuity, which pays annuities beginning from the year of age 66 until death, will change from \$13.94 to \$16.70 per \$1. Even a small amount of change for the price of the annuity is crucial for insurance companies and social security. Suppose the annual payments is \$20000 and there are 50000 individuals under cover, then every \$0.5 lower pricing will result in a \$500 million shortfall. Thus a better mortality forecasting, which guarantees a more accurately estimating of life expectancy and pricing of life

annuity, is crucial to reduce the social security risks.

This chapter aims to model and forecast the age-specific mortality data of the US population in the Human Mortality Database (HMD) (91) obtained in July 2018. After preprocessing, the annual age-specific death rates under study consist of a matrix data with 84 yearly observations (1933-2016) for 91 ages (0-90). Modelling and forecasting mortality data poses a challenge for traditional statistical analysis and multivariate time series analysis, as the dimension 91 is comparable to the sample size (or time length) 84. This high dimensional setting incurs curse of dimensionality. Dimension reduction is a remedy method that extracts representative features or patterns of available high dimensional data. Statistical analysis on extracted features and recovery of corresponding inference on original data are common techniques in high dimensional data analysis. However, optimal features for specific statistical inference are rarely studied. This chapter contributes to seeking linear features to attain optimal forecasting of the US mortality data. Roughly speaking, a linear feature is linear combination of annually death rates over total 91 ages, which is a univariate time series that summarized a 91-dimensional time series linearly. Before introducing the formal statistical model, we analyse the US mortality data qualitatively and interpret the features in pursuit intuitively.

We consider the logarithms of the death rates because this transformation makes positive-valued original data spread over total real-value set, which results in easier statistical modelling and losing no generality (Booth and Tickle [2008]). As illustration, Table 2.1 shows the structure of the historical log death rates as well as the purpose of forecasting. It demonstrates a classical problem: modelling and forecasting a high dimensional time series. Through two descriptive graphs we investigate the characteristics of the 91 time series under study. While Figure 2.2 exhibits relations of mortality data for all ages at different years, Figure 2.1 illustrates time-serial trend for each age. US mortality data possess quite systematically distinct characteristics. Firstly, the death rates are decreasing over

the years in common. Secondly, similar curves from different years in Figure 2.2 state that mortality at different ages has strong relations, especially consecutive ages. These crucial points create the opportunity of utilizing common information among ages in forecasting. More specifically, we can consider the features that represent the common time-serial trend at most to benefit the forecasting and the features that capture the common variations to help improve the model fitting. As two types of features are different in the sense of representing different characteristics of the mortality data, we will build a two-style factor model to capture those features. In fact, linear features with most variation help improve model fitting while those with larger auto-covariance enhance modelling efficiency, which play significant roles in accurate forecasting.

Table 2.1: The Log Central Death Rates of the US

	Historical data					Forecasts		
	1933	1934	1935	...	2016	2017	2018	...
0	−2.792	−2.681	−2.789	...	...	?	?	?
1	−4.661	−4.551	−4.720	...	...	?	?	?
2	−5.437	−5.328	−5.486	...	...	?	?	?
3	−5.775	−5.735	−5.816	...	...	?	?	?
4	−6.038	−6.011	−6.031	...	...	?	?	?
5	−6.227	−6.200	−6.210	...	...	?	?	?
...	...	...	...	...	...	?	?	?
90+	...	...	...	...	...	?	?	?

A large body of literature study diverse models on mortality forecasting. A detailed review is provided in Booth and Tickle [2008]. One seminal paper on US mortality forecasting is Lee and Carter [1992]. Lee-Carter model is the most prominent method for forecasting mortality rates, and it is used by the US Bureau of the Census as the benchmark model (Hollmann et al. [1999]). Lee-Carter model is also from a dimension-reduction point of view, which first extracts the common features of mortality for all ages, then makes use of these common features' forecasts to recover forecasting on mortality data. By utilizing principal component analysis (PCA), Lee-Carter model pursues common features that re-

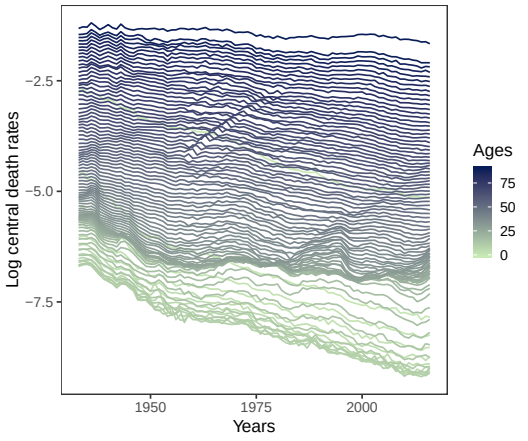


Figure 2.1: The Log Central Death Rates, years $1933 - 2016$ for ages 0 to 90+.

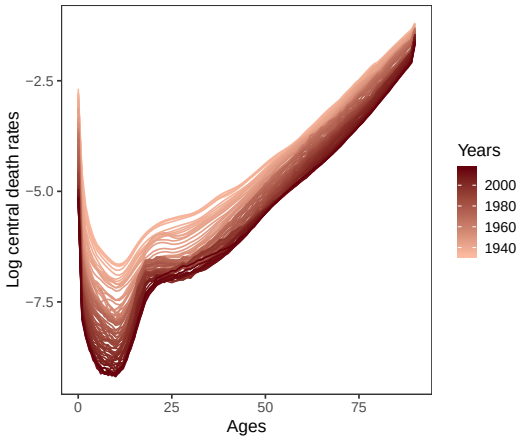


Figure 2.2: The Log Central Death Rates, ages 0 to 90+ for years $1933 - 2016$.

tain the most variation of mortality data. As a popular dimension reduction technique, PCA can be traced to that of Anderson [2003]; Jolliffe [2002]). Differently from the static PCA (the standard PCA used in the Lee-Carter model), several papers search for common features that drive the time-serial dependence of the original high dimensional time series. Brillinger [1975] and Hörmann et al. [2015] extended the static PCA to dynamic PCA, which extract features from a Fourier transformation on covariance and auto-covariances with different time-lags. Lam et al. [2011] and Chang et al. [2018] extracted dynamic features by assembling auto-covariances in another way while excluding the covariance.

As analyzed above, linear features represent common variation and common temporal trend should be different. Based on this point, we propose a two-step dimension reduction approach to seek these two kinds of features, both of which play significant roles in accurate forecasting. We decompose mortality data into three parts: a strong dynamic part driven by a lower-dimensional factor time series; a weak dynamic but strong variation part represented by another lower-dimensional factor time series; and an error part that is a high dimensional time series with weak serial dependence as well as small variation. The description of the error part illustrates the aim of such modelling is to capture dynamic

common features and static features as most as possible. Estimation of this two-type factor model is carried out by an eigenanalysis for an auto-covariance matrix and a covariance matrix, respectively. Through investigating asymptotic properties of the proposed method, the estimating for two kinds of factors have equally fast rates of convergence.

We show that our proposed method is capable to improve the forecasting of the US mortality rates compared to the benchmark methods. Moreover, from the application for estimating life expectancy and pricing the life annuity, we find that Lee-Carter method tend to lower estimate those value while our method provides close result to the true value, when using the data of 1933 to 1986 as training set and those of 1987 to 2016 as test set. Using the same training and test sets, Lee-Carter methods tends to price lower for around \$0.2 to \$0.4 per \$1 annual payment, which is indeed a big risk for social security considering the large amount of payments and population under cover in the real world. On the other hand, the price from our method is about \$0.02 higher per \$1, which is a remarkable improvement.

The rest of the chapter is organized as follows. In Section 2.2, the details of the model are described, including the estimation and forecasting method. We also provide a practical algorithm of the proposed method in Section 2.2. The asymptotic properties of the proposed method are presented in Section 2.4 and the corresponding proof is in Appendix A.2. In Section 2.3, we discuss the relationship and difference of our proposed method with static PCA and dynamic PCA. We present simulations of those difference in Section 2.5. Also in Section 2.5, we show the out of sample forecasting performance of our method compared to conventional methods with several examples. In Section 2.6, we compare the forecasting performance of our method with conventional methods on the US age-specific mortality rates. Finally, We apply our method to do a long term forecasting on the US population and comparing the computed future remaining life expectancy and the present value of life annuity with those obtained by Lee-

Carter model. The conclusion is presented in Section 2.7.

The notations in this chapter are summarized here. For an $p \times n$ matrix $\boldsymbol{C}$, we denotes its transpose as $\boldsymbol{C}^\top$, the square root of the maximum eigenvalue of $\boldsymbol{C}\boldsymbol{C}^\top$ as $\|\boldsymbol{C}\|$, and the square root of the smallest nonzero eigenvalue of $\boldsymbol{C}\boldsymbol{C}^\top$ as $\|\boldsymbol{C}\|_{\min}$. For a $k \times k$ matrix $\boldsymbol{F}$, $\lambda_i(\boldsymbol{F})$ indicates the i-th largest eigenvalue of the matrix $\boldsymbol{F}$. For a non-symmetric matrix $\boldsymbol{S}$, we use $\sigma_j(\boldsymbol{S})$ to denote the singular value of the matrix $\boldsymbol{S}$, which corresponds to the j-th largest eigenvalue of the matrix $\boldsymbol{S}\boldsymbol{S}^\top$. $\boldsymbol{I}_p$ represents $p-$dimensional identity matrix. All vectors are column vectors. The notation $a \asymp b$ means that $a = O(b)$ and $b = O(a)$. $\xrightarrow{i.p.}$ denotes convergence in probability. We use $P, T \to \infty$ to denote that P and T go to infinity jointly.

2.2 Model and estimation

Let $\boldsymbol{m}_t = (m_{1,t}, m_{2,t}, \ldots, m_{P,t})^\top$ be the US age-specific death rates in year t, where $m_{p,t}$ is the death rate for age p in year t with $p = 1, 2, \ldots, P$ and $t = 1, 2, \ldots, T$. The historical mortality data is available annually from the year 1933 to the year 2016 for ages from 0 to 90+. For high dimensional time series $\{\boldsymbol{m}_t, t = 1, 2, \ldots, T\}$, the time-serial length and the dimension are $T = 84$ and $P = 91$, respectively. We will propose a two-step dimension reduction model on the log transformation of $\boldsymbol{m}_t$ which is denoted by $\boldsymbol{y}_t = (\ln(m_{1,t}), \ln(m_{2,t}), \ldots, \ln(m_{P,t}))^\top$. It is worth noting that building a model is much easier on $\boldsymbol{y}_t$ than that on $\boldsymbol{m}_t$ because $\boldsymbol{m}_t$ take non-negative values.

In this section, we will first introduce a two-step dimension reduction model on the historical mortality data. Secondly a forecasting procedure based on extracted features will be provided.

2.2.1 Two-style factor model

As analyzed in last section, death rates for all 91 ages possess common features that drive common time-serial trend and common variation, respectively. This leads us to the following two-style factor model: for any $t = 1, 2, \ldots, T$ $(T = 84)$,

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{k}_t^{(1)} + \boldsymbol{u}_t, \tag{2.2.1}$$

$$\boldsymbol{u}_t = \boldsymbol{A}\boldsymbol{k}_t^{(2)} + \boldsymbol{\varepsilon}_t, \tag{2.2.2}$$

where $\boldsymbol{k}_t^{(1)} = \left(k_{1t}^{(1)}, k_{2t}^{(1)}, \ldots, k_{r_1 t}^{(1)}\right)^\top$ is an $r_1 \times 1$ latent process with $r_1 < P$, which represents common temporal trends; $\boldsymbol{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{r_1})$ is a $P \times r_1$ unknown deterministic coefficients matrix; similarly, $\boldsymbol{k}_t^{(2)} = \left(k_{1t}^{(2)}, k_{2t}^{(2)}, \ldots, k_{r_2 t}^{(2)}\right)^\top$ is an $r_2 \times 1$ latent process with $r_2 < P$, which indicates common variation among all ages; $\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{r_2})$ is the corresponding $P \times r_2$ unknown deterministic coefficients matrix; and $\boldsymbol{\varepsilon}_t$ is an error component.

Here we assume r_1 and r_2 are both unknown positive integers. Once P is much larger than $(r_1 + r_2)$, an effective dimension reduction is achieved because the original time series $\boldsymbol{y}_t$ is driven by a much lower dimensional time series $\left(\boldsymbol{k}_t^{(1)}, \boldsymbol{k}_t^{(2)}\right)$. We also call model (2.2.1) and (2.2.2) the factor models with $\boldsymbol{k}_t^{(1)}$ and $\boldsymbol{k}_t^{(2)}$ being common factors, respectively. And $\boldsymbol{B}$ and $\boldsymbol{A}$ are the corresponding factor loadings, respectively. Factor model is a popular dimension reduction model in high dimensional statistics, which is investigated in huge amounts of literature including Lam and Yao [2012], Lam et al. [2011], Bai [2002].

It is noted that this two-style factor model involves two kinds of common factors $\boldsymbol{k}_t^{(1)}$ and $\boldsymbol{k}_t^{(2)}$, which represent common temporal trends and common variation among all p ages, respectively. These two kinds of common factors are necessary in producing good forecasting results.

Because all elements in the model are unknown, including factor loadings and common factors, we should impose identification conditions to make the model

well-defined. First, we assume that the rank of factor loadings $\boldsymbol{B}$ and $\boldsymbol{A}$ are equal to r_1 and r_2, respectively. Otherwise, the two parts $\boldsymbol{B}\boldsymbol{k}_t^{(1)}$ and $\boldsymbol{A}\boldsymbol{k}_t^{(2)}$ can be represented in terms of factor models with even lower dimension. Moreover, as factors and factor loadings are all unknown, for any $r_1 \times r_1$ invertible matrix $\boldsymbol{H}$, if we substitute the factor model part $\left(\boldsymbol{B}, \boldsymbol{k}_t^{(1)}\right)$ with $\left(\boldsymbol{B}\boldsymbol{H}, \boldsymbol{H}^{-1}\boldsymbol{k}_t^{(1)}\right)$, the term $\boldsymbol{B}\boldsymbol{k}_t^{(1)}$ is unchanged. It is also true for the term $\boldsymbol{A}\boldsymbol{k}_t^{(2)}$. To avoid such matters, we impose the following assumption.

Assumption 2.1. *Orthogonal Condition.* $\boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{I}_{r_1}$, $\boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I}_{r_2}$, *where* $\boldsymbol{I}_{r_1}$ *and* $\boldsymbol{I}_{r_2}$ *are* $r_1 \times r_1$ *and* $r_2 \times r_2$ *identity matrices, respectively.*

Under Assumption 2.1, the factor loading $\boldsymbol{B}$ and the common factor $\boldsymbol{k}_t^{(1)}$ are determined up to an orthogonal matrix $\boldsymbol{H}$, and the same for the pair $\left(\boldsymbol{A}, \boldsymbol{k}_t^{(2)}\right)$. In this way, Assumption 2.1 provides identification conditions between common factors and the corresponding factor loadings. It is also a common identification condition for factor models used in literature including Bai [2002], Lam and Yao [2012].

Secondly, we consider the identification between the two kinds of factor models. As mentioned earlier, the two kinds of factor parts represent common temporal trends and common variation of the data, respectively. Intuitively, we think the first common factor part possess stronger time serial dependence than the second factor part. Formally, we use the auto-covariance to distinguish the two parts, which is reasonable since auto-covariance can describe strength of time-serial dependence. Before introducing Assumption 2.2, we specify some notations. $\boldsymbol{\Sigma}_k^{(1)}(\ell) := cov\left(\boldsymbol{k}_t^{(1)}, \boldsymbol{k}_{t+\ell}^{(1)}\right)$ and $\boldsymbol{\Sigma}_k^{(2)}(\ell) := cov\left(\boldsymbol{k}_t^{(2)}, \boldsymbol{k}_{t+\ell}^{(2)}\right)$ are auto-covariance of $\boldsymbol{k}_t^{(1)}$ with lag ℓ and that of $\boldsymbol{k}_t^{(2)}$ with lag ℓ, respectively. For any matrix $\boldsymbol{C}$, let $||\boldsymbol{C}||$ be the square root of the maximum eigenvalue of $\boldsymbol{C}\boldsymbol{C}^\top$ and $||\boldsymbol{C}||_{\min}$ be the square root of the smallest nonzero eigenvalue of $\boldsymbol{C}\boldsymbol{C}^\top$.

Assumption 2.2. *Identification between* $\boldsymbol{k}_t^{(1)}$ *and* $\boldsymbol{k}_t^{(2)}$. $\left\|\boldsymbol{\Sigma}_k^{(1)}(\ell)\right\| \asymp P^{1-\delta_1} \asymp \left\|\boldsymbol{\Sigma}_k^{(1)}(\ell)\right\|_{\min}$, $\left\|\boldsymbol{\Sigma}_k^{(2)}(\ell)\right\| \asymp P^{1-\delta_2} \asymp \left\|\boldsymbol{\Sigma}_k^{(2)}(\ell)\right\|_{\min}$, *where* $0 \le \delta_1 < \delta_2 \le 1$.

Assumption 2.2 imposes different orders for eigenvalues of the auto-covariance matrices for the two kinds of common factors. The order $P^{1-\delta_1}$ for $\boldsymbol{k}_t^{(1)}$ is larger than the order $P^{1-\delta_2}$ for $\boldsymbol{k}_t^{(2)}$ as $\delta_1 < \delta_2$, which ensures that the time-serial dependence of $\boldsymbol{k}_t^{(1)}$ is stronger than that of $\boldsymbol{k}_t^{(2)}$. In view of this, Assumption 2.2 identifies the two kinds of factor models via their time-serial dependence. In other words, the first factor model part extracts common factors with stronger time-serial dependence, which also takes higher priority in the forecasting. This kind of identification condition is utilized in Lam and Yao [2012].

Thirdly, we distinguish the second kind of factor part from the error component of the model. After extracting the common temporal trends in the first part of the model, the aim on better forecasting stimulate us to pursue further necessary features in the data. Compared with the factor $\boldsymbol{k}_t^{(1)}$, the factor part $\boldsymbol{k}_t^{(2)}$ has weaker time-serial dependence. It has little interest for forecasting improvement. However, it implies large amounts of common variation of the data. Neglecting it will result in bad model fitting of the original data. As better model fitting also plays an important role in forecasting improvement, we would like to keep them in the dimension reduction as well.

Assumption 2.3. *Identification between $\boldsymbol{k}_t^{(2)}$ and $\boldsymbol{\varepsilon}_t$.*
$\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{k}_t^{(2)} \boldsymbol{k}_t^{(2)\top} \xrightarrow{i.p.} \boldsymbol{\Sigma}_k^{(2)}(0) > 0$, *as* $P, T \to \infty$. *Here* $\boldsymbol{\Sigma}_k^{(2)}(0)$ *is a deterministic* $r_2 \times r_2$ *positive definite matrix.*

Assumption 2.3 is a common condition on factor model in the sense of the factors representing most variation of the data. It is used in Bai [2002].

At last, we impose some conditions on the error component.

Assumption 2.4. *Error components.*

1. $\mathbb{E}(\varepsilon_{it}) = 0$. $\{\boldsymbol{\varepsilon}_t : t \geq 1\}$ *is strictly stationary.*
2. $\sum_{i=1}^{P}\sum_{j=1}^{P}\sum_{t=1}^{T}\sum_{s=1}^{T} |\mathbb{E}(\varepsilon_{it}\varepsilon_{js})| = O(NT)$ *and* $\sum_{i=1}^{P}\sum_{j\neq i} |\sigma_{\varepsilon,ij}| = O(P)$, *where* $\sigma_{\varepsilon,ij} := \mathbb{E}(\varepsilon_{it}\varepsilon_{jt})$.

Condition (2) of Assumption 2.4 ensure that weak cross-sectional dependence and time-serial dependence in the error component. This condition indicates that no obvious common variation and common temporal trends involved in the error component.

In summary, Assumptions 2.1-2.4 create a well-defined two-style factor model (2.2.1) and (2.2.2). Next, we will consider how to estimate the two kinds of common factors for further forecasting.

2.2.2 Estimation approach

Based on the identification between $\boldsymbol{k}_t^{(1)}$ and $\boldsymbol{k}_t^{(2)}$, the factor $\boldsymbol{k}_t^{(1)}$ will play a leading role in the auto-covariance matrix $\boldsymbol{\Sigma}_y(\ell) := cov\left(\boldsymbol{y}_t, \boldsymbol{y}_{t+\ell}\right)$ with ℓ being a positive integer. To see this point clearly, we do some calculation under the case of $\boldsymbol{k}_t^{(1)}$ and $\boldsymbol{k}_t^{(2)}$ being independent. Then it follows from (2.2.1) and (2.2.2) that

$$\boldsymbol{\Sigma}_y(\ell) = \boldsymbol{B}\boldsymbol{\Sigma}_k^{(1)}(\ell)\boldsymbol{B}^\top + \boldsymbol{A}\boldsymbol{\Sigma}_k^{(2)}(\ell)\boldsymbol{A}^\top + \boldsymbol{\Sigma}_\varepsilon(\ell), \tag{2.2.3}$$

where $\boldsymbol{\Sigma}_\varepsilon(\ell) = cov\left(\boldsymbol{\varepsilon}_t, \boldsymbol{\varepsilon}_{t+\ell}\right)$.

With Assumption 2.2 and Assumption 2.4, $\boldsymbol{B}\boldsymbol{\Sigma}_k^{(1)}(\ell)\boldsymbol{B}^\top$ is the leading term of $\boldsymbol{\Sigma}_y(\ell)$ in the sense of spectral norm. As auto-covariance matrices are not symmetric, we consider the matrix

$$\boldsymbol{L}(\ell) := \boldsymbol{\Sigma}_y(\ell)\boldsymbol{\Sigma}_y(\ell)^\top. \tag{2.2.4}$$

It is easy to obtain that the columns of $\boldsymbol{B}$ are the eigenvectors of the matrix $\boldsymbol{B}\boldsymbol{\Sigma}_k^{(1)}(\ell)\boldsymbol{\Sigma}_k^{(1)}(\ell)^\top\boldsymbol{B}^\top$ corresponding to its non-zero eigenvalues. In fact, if $\boldsymbol{C} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{P-r_1})$ is a $P \times (P - r_1)$ matrix for which $(\boldsymbol{B}, \boldsymbol{C})$ forms a $P \times P$ orthogonal matrix, that is $\boldsymbol{C}^\top\boldsymbol{B} = \boldsymbol{0}$ and $\boldsymbol{C}^\top\boldsymbol{C} = \boldsymbol{I}_{P-r_1}$, then we have $\left(\boldsymbol{B}\boldsymbol{\Sigma}_k^{(1)}(\ell)\boldsymbol{\Sigma}_k^{(1)}(\ell)^\top\boldsymbol{B}^\top\right)\boldsymbol{C} = \boldsymbol{0}$. That is, the columns of $\boldsymbol{C}$ are eigenvectors of $\boldsymbol{B}\boldsymbol{\Sigma}_k^{(1)}(\ell)\boldsymbol{\Sigma}_k^{(1)}(\ell)^\top\boldsymbol{B}^\top$ corresponding to zero eigenvalues.

Furthermore, the matrix $\boldsymbol{B}\boldsymbol{\Sigma}_k^{(1)}(\ell)\boldsymbol{\Sigma}_k^{(1)}(\ell)^\top\boldsymbol{B}^\top$ is the leading term of the matrix $\boldsymbol{L}(\ell)$ in the sense of spectral norm. Hence the columns of $\boldsymbol{B}$ are close to the eigenvectors of the matrix $\boldsymbol{L}(\ell)$ corresponding to non-zero eigenvalues, approximately.

In terms of the analysis above, the eigendecomposition of $\boldsymbol{L}(\ell)$ provides a recovery method of the factor loading $\boldsymbol{B}$. Note that we use $\ell = 1$ in the estimation step, because the estimation of $\boldsymbol{B}$ is not sensitive to ℓ and the correlation is often at its strongest at the small time lag (Lam and Yao [2012]). Besides, after analyzing the US mortality data, we also find $\ell = 1$ is enough for the forecasting.

Back to the two-style factor model (2.2.1) and (2.2.2), given an estimator for the first factor part, the recovery of the second factor part is more straightforward. In fact, the model is reduced into a simpler form

$$\boldsymbol{y}_t - \boldsymbol{B}\boldsymbol{k}_t^{(1)} = \boldsymbol{A}\boldsymbol{k}_t^{(2)} + \boldsymbol{\varepsilon}_t. \tag{2.2.5}$$

Under Assumption 2.3, (2.2.5) is a classical factor model which can be estimated by the standard (static) PCA. See Fan et al. [2013].

In summary, the proposed novel dimension reduction method has two steps. The first step is to extract useful features which has good forecasting behaviors by a dynamic PCA procedure. The second step is to extract features, that retain the variations for each age by performing static PCA. After the dimension reduction, we get two sets of features, with which we would like to recover $\boldsymbol{y}_t$ as follows:

$$\widetilde{\boldsymbol{y}}_t = \sum_{i=1}^{r_1} \boldsymbol{b}_i k_{it}^{(1)} + \sum_{i=1}^{r_2} \boldsymbol{a}_i k_{it}^{(2)}, \tag{2.2.6}$$

where $k_{it}^{(1)}$ is the i^{th} feature extracted in the first step, which is a linear combination of the log scale age-specific death rates at time t, $k_{it}^{(2)}$ is that in the second step, $r_1 > 0$ and $r_2 > 0$ are the number of features extracted in two steps respectively which satisfy $r_1 + r_2 < P$, and $\boldsymbol{b}_i : P \times 1$, $\boldsymbol{a}_i : P \times 1$ are the

corresponding coefficients of the two sets of linear combinations. Hence, $\widetilde{\boldsymbol{Y}} = [\widetilde{\boldsymbol{y}}_1, \widetilde{\boldsymbol{y}}_2, \ldots, \widetilde{\boldsymbol{y}}_T]$ is a *low-dimensional representation* of $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T]$, as $\text{rank}\{\widetilde{\boldsymbol{Y}}\} \leq r_1 + r_2 < P$. Next we describe how we can get this low-dimensional representation 2.2.6 by the two-step dimension reduction in details.

The first step

Firstly, we assume that $\{\boldsymbol{y}_t\}_{t=1,2,\ldots,T}$ is covariance stationary and consider the following matrix

$$\boldsymbol{L}_1 = \boldsymbol{\Sigma}_{\boldsymbol{y}}(1)\boldsymbol{\Sigma}_{\boldsymbol{y}}(1)^\top,$$

where $\boldsymbol{\Sigma}_{\boldsymbol{y}}(1) = \text{cov}(\boldsymbol{y}_t, \boldsymbol{y}_{t-1})$. As $\boldsymbol{L}_1$ is a symmetric matrix, it can be decomposed as $\boldsymbol{L}_1 = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$. $P \times P$ matrix $\boldsymbol{Q}$ consists of the orthogonal eigenvectors of $\boldsymbol{L}_1$ in the columns and the columns are arranged such that the corresponding eigenvalues are in descending order. $\boldsymbol{\Lambda}$ is a $P \times P$ diagonal matrix with eigenvalues of $\boldsymbol{L}_1$ as the diagonal elements in descending order. As $\boldsymbol{Q}$ is an orthogonal matrix, we have $\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}$. Let $\boldsymbol{\mu}_{\boldsymbol{y}} = \mathbb{E}(\boldsymbol{y}_t)$, then we have

$$\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}} = \boldsymbol{Q}\boldsymbol{Q}^\top(\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}}).$$

By some simple rearrangement and let $\boldsymbol{b}_i$ be the i^{th} column of of $\boldsymbol{Q}$, which is the eigenvector corresponding to the i^{th} largest eigenvalue of $\boldsymbol{L}_1$, we have

$$\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}} = \sum_{i=1}^{r_1} \boldsymbol{b}_i\boldsymbol{b}_i^\top(\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}}) + \sum_{i=r_1+1}^{P} \boldsymbol{b}_i\boldsymbol{b}_i^\top(\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}}). \tag{2.2.7}$$

Without loss of generality, we assume $\boldsymbol{\mu}_{\boldsymbol{y}} = \boldsymbol{0}$ in the following analysis. Let $k_{it}^{(1)} = \boldsymbol{b}_i^\top(\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}}) = \boldsymbol{b}_i^\top\boldsymbol{y}_t$, and $\boldsymbol{u}_t = \sum_{i=r_1+1}^{P} \boldsymbol{b}_i\boldsymbol{b}_i^\top(\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}}) = \sum_{i=r_1+1}^{P} \boldsymbol{b}_i k_{it}^{(1)}$.

Then we can rewrite equation 2.2.7 as

$$\boldsymbol{y}_t = \sum_{i=1}^{r_1} \boldsymbol{b}_i k_{it}^{(1)} + \boldsymbol{u}_t. \tag{2.2.8}$$

Then the linear combination $k_{it}^{(1)}, i = 1, 2, \ldots, r_1$, are the features representing the time-serial trend, which have good forecasting behaviors.

The second step

The second step is equivalent to do a static PCA on $\boldsymbol{u}_t$ in 2.2.8. Let $\boldsymbol{\Sigma_u}(0) = var(\boldsymbol{u}_t)$, then the desired matrix for the second step is:

$$\boldsymbol{L}_2 = \boldsymbol{\Sigma_u}(0)\boldsymbol{\Sigma_u}(0)^\top.$$

Conducting eigen-decomposition on $\boldsymbol{L}_2$ and let $\boldsymbol{a}_i$ be the eigenvector corresponding to the i^{th} largest eigenvalue of $\boldsymbol{L}_2$. Then similar to that in the first step, $k_{it}^{(2)} = \boldsymbol{a}_i^\top \boldsymbol{u}_t$ is the j^{th} feature extracted from the second step, which captures the common variation. Then $\boldsymbol{u}_t$ can be expressed as:

$$\boldsymbol{u}_t = \sum_{i=1}^{r_2} \boldsymbol{a}_i k_{it}^{(2)} + \boldsymbol{\varepsilon}_t, \tag{2.2.9}$$

where $\boldsymbol{\varepsilon}_t = \sum_{i=r_2+1}^{P} \boldsymbol{a}_i k_{it}^{(2)}$. We can choose a $r_2 < (P - r_1)$ such that $\mathbb{E}(\|\boldsymbol{\varepsilon}_t^\top \boldsymbol{\varepsilon}_t\|)$ is small enough. Then $\sum_{i=1}^{r_2} \boldsymbol{a}_i k_{it}^{(2)}, t = 1, 2, \ldots, T$ is a low-dimensional representation of $\boldsymbol{u}_t, t = 1, 2, \ldots, T$.

Finally combining equation 2.2.8 and 2.2.9, we have:

$$\boldsymbol{y}_t = \sum_{i=1}^{r_1} \boldsymbol{b}_i k_{it}^{(1)} + \sum_{i=1}^{r_2} \boldsymbol{a}_i k_{it}^{(2)} + \boldsymbol{\varepsilon}_t,$$

and let

$$\widetilde{\boldsymbol{y}}_t = \sum_{i=1}^{r_1} \boldsymbol{b}_i k_{it}^{(1)} + \sum_{i=1}^{r_2} \boldsymbol{a}_i k_{it}^{(2)}, \tag{2.2.10}$$

which is a low-dimensional representation of $\boldsymbol{y}_t$ we get after the two-step dimension reduction procedure.

2.2.3 Forecasting

Recall that after the dimension reduction on the log scale death rate $\boldsymbol{y}_t$, we get a low-dimensional representation 2.2.10. Following Lee and Carter [1992], we can forecast $\boldsymbol{y}_{T+h}$ by forecasting the features $k_{i,T+h}^{(1)}$ and $k_{i,T+h}^{(2)}$first. In order to get the forecasts $\widehat{k}_{i,T+h}^{(1)}$ and $\widehat{k}_{i,T+h}^{(2)}$, we model $\{k_{it}^{(1)} : i = 1, 2, \ldots, r_1\}_{t=1,2,\ldots,T}$ and $\{k_{it}^{(2)} : i = 1, 2, \ldots, r_2\}_{t=1,2,\ldots,T}$ with standard time sires models and conduct h-step ahead forecasting with the models. Then together with 2.2.10, the h-step ahead forecasting for $\boldsymbol{y}_{T+h}$ is

$$\widetilde{\boldsymbol{y}}_{T+h} = \sum_{i=1}^{r_1} \boldsymbol{b}_i \widehat{k}_{i,T+h}^{(1)} + \sum_{i=1}^{r_2} \boldsymbol{a}_i \widehat{k}_{i,T+h}^{(2)},$$

where $\widehat{k}_{i,T+h}^{(1)}$ and $\widehat{k}_{i,T+h}^{(2)}$ are predicted values of the features in h years after time T, $h = 1, 2, \ldots$.

Consequently, instead of conducting P forecasting models, we only need $\widehat{r}_1 + \widehat{r}_2 < P$ forecasting models. In our simulation and application on the US mortality data, we choose $ARIMA(p, d, q)$ model to forecast the time series, and we use BIC to choose the parameters p, d, q for each model.

2.2.4 Practical algorithm

The practical procedure for the dimension reduction and forecasting is summarized in Algorithm 1.

Algorithm 1: Data-adaptive Dimension Reduction for Mortality Forecasting

Input: Data $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T] \in \mathbb{R}^{P \times T}$; Desired rank $\leq P$.

Output: Low-dimensional representation of $\boldsymbol{Y}$; $h-$steps ahead forecasts of $\boldsymbol{y}_T$, h = 1,2, ...

Dimension Reduction Step 1:

Compute the sample mean $\overline{\boldsymbol{y}} = T^{-1} \sum_{t=1}^{T} \boldsymbol{y}_t$;

Compute the sample auto-covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1) = \frac{1}{T-1} \sum_{t=1}^{T-1} (\boldsymbol{y}_{t+1} - \overline{\boldsymbol{y}})(\boldsymbol{y}_t - \overline{\boldsymbol{y}})^\top$;

Compute sample matrix for the first step $\widehat{\boldsymbol{L}}_1 = \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1) \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1)^\top$;

Conduct eigendecomposition on $\widehat{\boldsymbol{L}}_1$ and get $\widehat{\boldsymbol{b}}_1, \ldots, \widehat{\boldsymbol{b}}_{\widehat{r_1}}$, the eigenvectors corresponding to the largest $\widehat{r_1}$ eigenvalues of $\widehat{\boldsymbol{L}}_1$;

Compute the first sets of features $\widehat{k}_{it}^{(1)} = \widehat{\boldsymbol{b}}_i^\top (\boldsymbol{y}_t - \overline{\boldsymbol{y}}), i = 1, \ldots, \widehat{r_1}, t = 1, \ldots, T$;

Dimension Reduction Step 2:

Compute $\widehat{\boldsymbol{u}}_t = (\boldsymbol{y}_t - \overline{\boldsymbol{y}}) - \sum_{i=1}^{\widehat{r_1}} \widehat{\boldsymbol{b}}_i \widehat{k}_{it}^{(1)}$;

Compute sample the covariance matrix of $\widehat{\boldsymbol{u}}_t$, $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}(0) = \frac{1}{T} \sum_{t=1}^{T} \widehat{\boldsymbol{u}}_t \widehat{\boldsymbol{u}}_t^\top$;

Compute sample matrix for second step $\widehat{\boldsymbol{L}}_2 = \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\varepsilon}}(0) \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\varepsilon}}(0)^\top$;

Conduct eigendecomposition on $\widehat{\boldsymbol{L}}_2$ and get $\widehat{\boldsymbol{a}}_1, \ldots, \widehat{\boldsymbol{a}}_{\widehat{r_2}}$, the eigenvectors corresponding to the largest $\widehat{r_2}$ eigenvalues of $\widehat{\boldsymbol{L}}_2$;

Compute the second sets of features $\widehat{k}_{it}^{(2)} = \widehat{\boldsymbol{a}}_i^\top \widehat{\boldsymbol{u}}_t, i = 1, \ldots, \widehat{r_1}, t = 1, \ldots, T$;

Compute $\widehat{\boldsymbol{y}}_t = \overline{\boldsymbol{y}} + \sum_{i=1}^{\widehat{r_1}} \widehat{\boldsymbol{b}}_i \widehat{k}_{it}^{(1)} + \sum_{i=1}^{\widehat{r_2}} \widehat{\boldsymbol{a}}_i \widehat{k}_{it}^{(2)}, t = 1, \ldots, T$, and the estimated low-dimensional representation of $\boldsymbol{Y}$ is $\widehat{\boldsymbol{Y}} = [\widehat{\boldsymbol{y}}_1, \ldots, \widehat{\boldsymbol{y}}_T]$;

Forecasting Step:

Fit $\widehat{k}_{it}^{(1)}, i = 1, \ldots, \widehat{r_1}, t = 1, \ldots, T$ and $\widehat{k}_{it}^{(2)}, i = 1, \ldots, \widehat{r_2}, t = 1, \ldots, T$ with standard ARIMA(p,d,q) models respectively;

Compute $\widehat{k}_{i,T+h}^{(1)}, \widehat{k}_{j,T+h}^{(2)}$, the $h-$step ahead forecasts of the features, with the fitted ARIMA models; $i = 1, \ldots, \widehat{r}_1$; $j = 1, \ldots, \widehat{r}_2$;

Compute the $h-$step ahead forecasts for $\boldsymbol{y}_T$ by $\widehat{\boldsymbol{y}}_{T+h} = \overline{\boldsymbol{y}} + \sum_{i=1}^{\widehat{r_1}} \widehat{\boldsymbol{b}}_i \widehat{k}_{i,T+h}^{(1)} + \sum_{i=1}^{\widehat{r_2}} \widehat{\boldsymbol{a}}_i \widehat{k}_{i,T+h}^{(2)}$.

In our simulation and analysis of the US mortality data, we estimate the value of both r_1 and r_2 by the criterion,

$$\widehat{r} = \underset{1\le i\le R}{\operatorname{argmin}} \frac{\widehat{\lambda}_{i+1}}{\widehat{\lambda}_i}, \tag{2.2.11}$$

where $\widehat{\lambda}_i, i = 1, 2, \ldots, R$ are the eigenvalues of $\widehat{\boldsymbol{L}}_1$ or $\widehat{\boldsymbol{L}}_2$ in descending order, and $\max(r_1, r_2) < R < P$. This criterion is justified in Lam and Yao [2012] and Ahn and Horensten [2013] for auto-covariance matrices and covariance matrices on high dimensional data, respectively.

As mentioned in Lam and Yao [2012], in practice, the parameter R is chosen as $\frac{1}{2}\min(P, T)$. It is worthy being mentioned that the number of nonzero eigenvalues of the sample matrix $\widehat{\boldsymbol{L}}_1$ and $\widehat{\boldsymbol{L}}_2$ is no larger than $\min(P, T)$.

2.3 Relationship with existing methods

The methods which forecast mortality via features' forecasting date back to the Lee-Carter model (60). For general comparison, consider the following one factor model

$$y_{x,t} = \ln(m_{x,t}) = a_x + b_x k_t + u_{x,t},$$

where a_x is a constant for each x, k_t is an unobserved time series (the feature summarizing the the original high-dimensional time series), b_x is the loading of the feature k_t to each age x, and $u_{x,t}$ is the error term. One can recover the $h-$step ahead forecasting of $y_{x,t}$ via the forecasting of k_t. Therefore, how we extract the feature k_t by dimension reduction is the main difference.

2.3.1 Static PCA method

The most popular method for mortality modelling, the Lee-Carter model, utilizes static PCA to estimate k_t. The extracted feature k_t, which performs the most important role in the forecasting, represents the principal component that explains the most of the variance of the original data. It, however, doesn't take the serial dependence along time for the original data into consideration. Mathematically, the estimation solves the following objective

$$\max_{\boldsymbol{b}} \operatorname{var}(\boldsymbol{b}^\top \boldsymbol{y}_t)$$

to get $\widetilde{k}_t = \widetilde{\boldsymbol{b}}^\top(\boldsymbol{y}_t - \bar{\boldsymbol{y}}), t = 1, \ldots, T$. This solution has the smallest average squared reconstruction error $\mathbb{E}(\|\boldsymbol{u}_t^\top \boldsymbol{u}\|^2)$ [Brillinger, 1975], while does not seem to have strong forecasting ability. For example, suppose $\boldsymbol{y}_t = (y_{1t}, y_{2t}, \ldots, y_{pt})^\top, t = 1, 2 \ldots, T$, with y_{1t} and y_{2t} have huge variances and very weak time serial dependence, while the rest have small variances but strong time serial dependence. Performing static PCA on $\boldsymbol{y}_t$ will get a principal component which puts most of the loading on y_{1t}, y_{2t}. The forecasting based on this principal component will heavily depend on the pattern of y_{1t}, y_{2t} and not make use of the strong serial dependence information of the rest, which may lead to a misleading forecasting.

On the other hand, the first step of our proposed method extracts k_t from the auto-covariance matrix, which contains sufficient time-serial dependence information. Thus it is expected to have stronger forecasting ability comparing to the feature extracted by the static PCA.

2.3.2 Dynamic PCA method

There are several dynamic PCA methods available to do the dimension reduction, which usually involve the auto-covariance matrices and also utilize the time-serial dependence information, for example, Lam et al. [2011], Hörmann et al. [2015],

Brillinger [1975], and Chang et al. [2018]. Those methods can be used to extract feature k_t as well. For comparison purpose with our method, we consider one of them described in the following.

Define $\boldsymbol{\Sigma}_{\boldsymbol{y}}(\ell) = \text{cov}(\boldsymbol{y}_t, \boldsymbol{y}_{t+\ell}), \ell = 0, 1, 2, \ldots$, and nonnegative definite matrix

$$\boldsymbol{L} = \sum_{\ell=0}^{\ell_0} \boldsymbol{\Sigma}_{\boldsymbol{y}}(\ell)\boldsymbol{\Sigma}_{\boldsymbol{y}}(\ell)^\top \tag{2.3.1}$$

With matrix 2.3.1, k_t can be estimated by $\widehat{k}_t = \widehat{\boldsymbol{b}}^\top(\boldsymbol{y}_t - \bar{\boldsymbol{y}}), t = 1, \ldots, T$, where $\widehat{\boldsymbol{b}}$ is the estimated eigenvector of the sample matrix $\widehat{\boldsymbol{L}}$ corresponding to its largest eigenvalue. This is similar to Hörmann et al. [2015] and Brillinger [1975], but assigns different weights on those covariances, while in Lam et al. [2011], it does not include $\boldsymbol{\Sigma}_{\boldsymbol{y}}(0)$. If $\ell_0 = 0$, it is the same with the static PCA. If $\ell_0 = 1$, $\boldsymbol{L}$ can be seen as a combination of the two steps in our method. If $\ell_0 > 1$, $\boldsymbol{L}$ aggregates more lagged covariances than our method.

There are similarities and advantages of our method compared with dynamic PCA. On one hand, the first step of our method is motivated by the dynamic PCA that makes use of the auto-covariance matrices to obtain forecasting ability for the features. While from the empirical and simulating studies, we find the lag 1 auto-covariance is enough for the mortality data and data with similar structure. To make the method simple and easy to apply, our method only involves the most useful auto-covariance. If the data structure changes, more information may need to be included. On the other hand, we intend to maximize the forecasting ability instead of balancing several characteristics of the features. The dynamic PCA provides only one set of features which balance the information of the temporal trend and the variation, while our proposed method extracts two sets of features via a step-wise procedure. The first sets of features represent the temporal trend, which benefit the forecasting, and the second parts capture variations which provides sufficient information for the recovering and also good for the forecasting. The features are linear combinations of the original data and

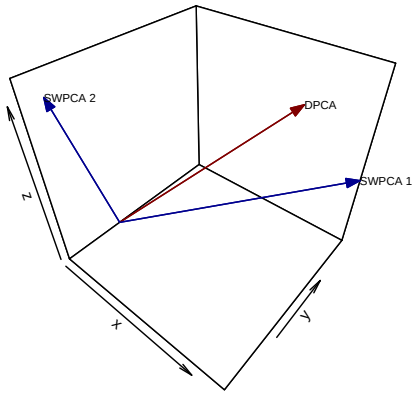


Figure 2.3: The directions of the features; A comparison between the DPCA and SWPCA.

the coefficients forming these linear combinations represent the directions which the original data is transformed to. Therefore, we can visualize the features by the directions. Figure 2.3 shows a simple example of the estimated directions for our method and the dynamic PCA. Data is generated the same as in *Example* 1 described in Section 2.5 with $P = 3$ and $T = 20$. The red arrow is the first direction for the dynamic PCA (DPCA), and the blue ones are those for the first step and the second step of our method (SWPCA), respectively. It is clear that they are different directions and the red one can be seen as a direction which makes a trade-off on the other two. In addition, if we take a detailed look at the matrix $\boldsymbol{L}$, we find that it is actually hard to tell and explain what information is contained in this matrix, as it is a mix of several auto-covariances and the variance matrix. Our method, on the other hand, is stepwise, thus it has a clear goal for each step.

2.4 Asymptotic properties

In this section, we establish the rates of convergence for the two-step estimators of the factor loadings. Additional to the Assumptions 2.1-2.4, we impose the following assumptions for the asymptotic theory.

Assumption 2.5. *Relation between the error and the factors.* $\boldsymbol{\varepsilon}_t$ *independent of* $\boldsymbol{k}_t^{(1)}$ *and* $\boldsymbol{k}_t^{(2)}$.

Remark 2.1. *For simplicity of techniques and without loss of generality, the Assumption 2.5 assumes independent relationship between the error component and the two kinds of factors.*

Assumption 2.6. *Relation between* $\boldsymbol{k}_t^{(1)}$ *and* $\boldsymbol{k}_t^{(2)}$. *Suppose that* $\left\|\boldsymbol{\Sigma}_k^{(21)}(\ell)\right\| \asymp \left\|\boldsymbol{\Sigma}_k^{(12)}(\ell)\right\|_{\min}$, $\left\|\boldsymbol{\Sigma}_k^{(12)}(\ell)\right\| = O\left(P^{1-\frac{\delta_2}{2}}\right)$, *where* $\boldsymbol{\Sigma}_k^{(21)}(\ell) = cov\left(\boldsymbol{k}_{t+\ell}^{(2)}, \boldsymbol{k}_t^{(1)}\right)$, $\boldsymbol{\Sigma}_k^{(12)}(\ell) = cov\left(\boldsymbol{k}_{t+\ell}^{(1)}, \boldsymbol{k}_t^{(2)}\right)$ *and* δ_2 *is defined in Assumption 2.2.*

Remark 2.2. *The order of the eigenvalues of* $\boldsymbol{\Sigma}_k^{(21)}(\ell)$ *is not specified in Assumption 2.6. The reason is that the information involved in* $\boldsymbol{\Sigma}_k^{(21)}$ *participate in the recovery of the factor* $\boldsymbol{k}_t^{(1)}$. *The order of* $\left\|\boldsymbol{\Sigma}_k^{12}(\ell)\right\|$ *is restricted in order to make it not involved in the leading term when recovering* $\boldsymbol{k}_t^{(1)}$.

Assumption 2.7. *Dimension Condition.* $\frac{P}{T} \to c \in (0, \infty)$.

Remark 2.3. *The setting of the dimension* P *and the sample size* T *being comparable is under consideration because the number of ages is comparable to the length of time series for US mortality data. Note that when* P *and* T *are on the same order, the estimators for the eigenvalues and the eigenvectors may be no longer consistent. See Lam and Yao [2012], [Ahn and Horenstein, 2013]. However, the ratio based estimators for* r_1 *and* r_2 *can still work well.*

Assumption 2.8. $\left\{\left(\boldsymbol{k}_t^{(1)}, \boldsymbol{k}_t^{(2)}, \boldsymbol{\varepsilon}_t\right) : t \geq 1\right\}$ *is strictly stationary with finite fourth moments.*

Theorem 2.1. *In addition to Assumptions 2.1 - 2.8, we assume that*

$$\frac{P^{1-\delta_1}}{T} = o(1), \quad as \quad P, T \to \infty. \tag{2.4.1}$$

Then we have the following convergent rates

$$\left\|\widehat{\boldsymbol{B}} - \boldsymbol{B}\right\| = O_p\left(\frac{1}{T^{1/2}}\right), \quad \left\|\widehat{\boldsymbol{A}} - \boldsymbol{A}\right\| = O_p\left(\frac{1}{T^{1/2}}\right). \tag{2.4.2}$$

Remark 2.4. *Two kinds of factors are both strong factors in the sense of auto-correlation and variance, respectively. It is reasonable to obtain fast rates of convergence for both of them. In view of this, our proposed two-step estimators have good statistical performance, which is an advantage for forecasting improvement. Based on identification condition between factors and factor loadings,* $\boldsymbol{B}$ *is determined up to an orthogonal matrix. Due to technical proofs (some techniques in Lemma A.3), the estimator* $\widehat{\boldsymbol{B}}$ *here is the estimator up to an identity matrix.*

2.5 Simulations

In this section, we use simulated data to illustrate the advantages of the two-step dimension reduction method. For descriptive convenience, we use "SWPCA" to represent our method, "CPCA" to represent the static PCA method which was described in Section 2.3, and "DPCA" to represent the dynamic PCA method described in Section 2.3 with $\ell_0 = 1$.

For all the three examples, we first examined the variance and serial dependence (lag 1 auto-covariance) of the first estimated factor by the three methods, respectively. Secondly, we evaluated the serial dependence and the variations remained in the error terms for the three methods. Finally, we compared the forecasting performance for the 1 step and 5 steps ahead forecasting with the root mean squared forecasting error (FRMSE).

We show that our method extracts the feature with the largest auto-covariance

and leave the least information in the error terms. As a result, the two-step dimension reduction method provides the best forecasting results for all the three examples. The details of the simulations are described in the rest of this section. More simulation studies can be find in the Appendix A.1.

2.5.1 Data generating processes

We generate three simulation examples according to the following two-factor model:

$$\boldsymbol{y}_t = \boldsymbol{b}k_t + \boldsymbol{a}w_t + \boldsymbol{\varepsilon}_t$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are two independent $P \times 1$ vectors with elements generated from a uniform distribution $U(0,1)$ and $\boldsymbol{\varepsilon}_t$ is a $P \times 1$ error term with elements independently generated from a normal distribution $N(0, 0.2^2)$. For all the three examples, $\{k_t\}_{t=1,2,\ldots,T}$ is generated from $AR(1)$ model with coefficient 0.8, while $\{w_t\}_{t=1,2,\ldots,T}$ are different for each example.

- For example 1, $\{w_t\}_{t=1,2,\ldots,T}$ is generated from $N(0,1)$, which indicates the series of w_t are independent;
- For example 2, we add time-serial dependence to the feature w_t, hence $\{w_t\}_{t=1,2,\ldots,T}$ is generated from $AR(1)$ model with coefficient 0.05;
- At last in example 3, we increase the dependence in the series of w_t and generate them from $AR(1)$ model with coefficient 0.2.

2.5.2 Performance evaluation criterion

Firstly, we show the variance and serial dependence (lag 1 auto-covariance) of the first estimated factor of the three methods, respectively. The variance and

serial dependence of the first estimated factor are computed as follows:

$$\text{Time variance}\left(\widehat{k}_t\right) = \frac{1}{T-1}\sum_{t=1}^{T}\left(\widehat{k}_t - \frac{1}{T}\sum_{j=1}^{T}\widehat{k}_j\right)^2,$$

$$\text{Time dependence}\left(\widehat{k}_t\right) = \frac{1}{T-2}\sum_{t=1}^{T-1}\left(\widehat{k}_t - \frac{1}{T}\sum_{j=1}^{T}\widehat{k}_j\right)\left(\widehat{k}_{t+1} - \frac{1}{T}\sum_{j=1}^{T}\widehat{k}_j\right),$$

where $\widehat{k}_t$ is the estimated first feature at time t. Especially, for our method, we compare the estimated first feature from the first step as it is the feature which intends to improve the forecasting power. Besides, we also report the sum of the aforementioned quantities:

$$\text{Mix}(\widehat{k}_t) = \text{Time variance}(\widehat{k}_t) + \text{Time dependence}(\widehat{k}_t).$$

Secondly, we investigate the dependence and the variation remained in the error terms as follows:

$$\text{Time variance}\left(\widehat{\boldsymbol{\varepsilon}}_{\cdot t}\right) = \frac{1}{P}\sum_{p=1}^{P}\left(\frac{1}{T-1}\sum_{t=1}^{T}\left(\widehat{\varepsilon}_{pt} - \frac{1}{T}\sum_{j=1}^{T}\widehat{\varepsilon}_{pj}\right)^2\right),$$

$$\text{Time dependence}\left(\widehat{\boldsymbol{\varepsilon}}_{\cdot t}\right) = \frac{1}{T(T-1)}\sum_{t_1=1}^{T}\sum_{t_2=1,t_2\neq t_1}^{T}\left|cov\left(\widehat{\boldsymbol{\varepsilon}}_{\cdot t_1}, \widehat{\boldsymbol{\varepsilon}}_{\cdot t_2}\right)\right|,$$

$$\text{Cross-sectional variance}\left(\widehat{\boldsymbol{\varepsilon}}_{p\cdot}\right) = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{P-1}\sum_{p=1}^{P}\left(\widehat{\varepsilon}_{pt} - \frac{1}{P}\sum_{j=1}^{P}\widehat{\varepsilon}_{jt}\right)^2\right),$$

$$\text{Cross-sectional dependence}\left(\widehat{\boldsymbol{\varepsilon}}_{p\cdot}\right) = \frac{1}{P(P-1)}\sum_{p_1=1}^{P}\sum_{p_2=1,p_2\neq p_1}^{P}\left|cov\left(\widehat{\boldsymbol{\varepsilon}}_{p_1\cdot}, \widehat{\boldsymbol{\varepsilon}}_{p_2\cdot}\right)\right|,$$

where $\widehat{\varepsilon}_{pt}$ is the the error term for age p at time t, $\widehat{\boldsymbol{\varepsilon}}_{\cdot t}$ is error terms for all ages

at time t, $\widehat{\boldsymbol{\varepsilon}}_{p\cdot}$ is the error terms across all time for age p, and

$$cov\left(\widehat{\boldsymbol{\varepsilon}}_{\cdot t_1}, \widehat{\boldsymbol{\varepsilon}}_{\cdot t_2}\right) = \frac{1}{P}\sum_{p=1}^{P}\left(\widehat{\varepsilon}_{pt_1} - \frac{1}{P}\sum_{j=1}^{P}\widehat{\varepsilon}_{jt_1}\right)\left(\widehat{\varepsilon}_{pt_2} - \frac{1}{P}\sum_{j=1}^{P}\widehat{\varepsilon}_{jt_2}\right),$$

$$cov\left(\widehat{\boldsymbol{\varepsilon}}_{p_1\cdot}, \widehat{\boldsymbol{\varepsilon}}_{p_2\cdot}\right) = \frac{1}{T}\sum_{t=1}^{T}\left(\widehat{\varepsilon}_{p_1t} - \frac{1}{T}\sum_{j=1}^{T}\widehat{\varepsilon}_{p_1j}\right)\left(\widehat{\varepsilon}_{p_2t} - \frac{1}{T}\sum_{j=1}^{T}\widehat{\varepsilon}_{p_2j}\right).$$

To evaluate the forecasting performance, we show the the 1 step and 5 steps ahead root mean squared error, which is computed by

$$\text{FRMSE}(h) = \left(\frac{\sum_{i=0}^{h-1}\|\widehat{\boldsymbol{y}}_{T-i} - \boldsymbol{y}_{T-i}\|_2^2}{hP}\right)^{1/2}$$

where $h = 1, 5$ (the forecasting length), $\widehat{\boldsymbol{y}}_{T-i}$ is obtained by forecasting with $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{T-h}\}$, and $\boldsymbol{y}_{T-i}$ is the true value in the forecasting horizon.

2.5.3 Simulation results

We try different sets of (P, T): $(50, 50)$, $(50, 100)$, $(100, 100)$, $(100, 200)$, $(200, 200)$, as we would like to evaluate the performance under the situations that P and T are comparable. The results are shown in Table 2.2 to Table 2.5.

From Table 2.2, we can see that the CPCA method provides feature $\widehat{k}_t$ with the largest variance, while the first step of our method (SWPCA) captures $\widehat{k}_t$ with the largest lag 1 auto-covariance, which summarizes most of the time serial dependence information of the original data. These simulated results corroborate the analysis in Section 2.3.

On the other hand, we can see that the DPCA method provides a $\widehat{k}_t$ with the largest sum of variance and lag 1 auto-covariance. Our method utilizes the same information with the DPCA, while we have two steps. When we compare the feature from our first step with that of the DPCA method, it is not surprising that DPCA one has larger $\text{Mix}(\widehat{k}_t)$. However, our second step provides features

Table 2.2: Variance and Dependence of $\widehat{k}_t$

(P,T)	Time variance ($\widehat{k}_t$)			Time dependence ($\widehat{k}_t$)			Mix ($\widehat{k}_t$)		
	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA
Example 1 (AR(1) 0.8 + N(0,1))									
$(50,50)$	**51.102**	51.008	48.750	29.015	29.437	**30.174**	80.117	**80.445**	78.923
$(50,100)$	**53.436**	53.330	51.577	31.843	32.304	**32.938**	85.279	**85.635**	84.515
$(100,100)$	**107.483**	107.263	103.799	64.119	65.070	**66.341**	171.601	**172.333**	170.139
$(100,200)$	**110.269**	110.037	107.091	67.532	68.517	**69.651**	177.801	**178.554**	176.742
$(200,200)$	**221.091**	220.619	214.682	135.760	137.762	**140.053**	356.851	**358.381**	354.735
Example 2 (AR(1) 0.8 + AR(1) 0.05)									
$(50,50)$	**51.000**	50.909	48.941	29.409	29.806	**30.429**	80.409	**80.715**	79.371
$(50,100)$	**53.085**	52.986	51.430	31.958	32.382	**32.940**	85.043	**85.368**	84.371
$(100,100)$	**107.666**	107.466	104.384	65.591	66.440	**67.541**	173.257	**173.906**	171.925
$(100,200)$	**110.047**	109.838	107.249	68.630	69.497	**70.471**	178.677	**179.335**	177.719
$(200,200)$	**221.705**	221.278	216.268	139.463	141.213	**143.102**	361.168	**362.492**	359.369
Example 3 (AR(1) 0.8 + AR(1) 0.2))									
$(50,50)$	**51.572**	51.498	50.033	31.392	31.685	**32.093**	82.964	**83.183**	82.126
$(50,100)$	**54.963**	54.889	53.942	35.265	35.556	**35.873**	90.228	**90.445**	89.815
$(100,100)$	**108.933**	108.778	106.789	69.386	69.996	**70.666**	178.319	**178.774**	177.455
$(100,200)$	**110.179**	110.024	108.342	71.475	72.086	**72.697**	181.655	**182.110**	181.039
$(200,200)$	**224.522**	224.200	220.766	147.018	148.273	**149.494**	371.541	**372.472**	370.260

that capture the remaining variance, which is a necessary supplement for the first step to ensures that the final sets of features provide good fitting to the original data.

From Table 2.3 and 2.4, we can see that our method always provides the error terms with the smallest time and cross-sectional variance and dependence. It shows that SWPCA can capture most of the time-serial dependence and variation information of all the ages among the three methods. The better model fitting performance of our method is supported by these results.

Table 2.3: Variance across Time and Sections of the error terms

	Time Variance ($\widehat{\varepsilon}_{\cdot t}$)			Cross-sectional Variance ($\widehat{\varepsilon}_{p\cdot}$)		
(P,T)	CPCA	DPCA	SWPCA	CPCA	DPCA	SWPCA
Example 1 (AR(1) 0.8 + N(0,1))						
$(50,50)$	0.141	0.142	**0.038**	0.145	0.146	**0.038**
$(50,100)$	0.146	0.146	**0.038**	0.148	0.151	**0.038**
$(100,100)$	0.147	0.147	**0.039**	0.151	0.153	**0.039**
$(100,200)$	0.151	0.151	**0.039**	0.154	0.156	**0.039**
$(200,200)$	0.152	0.152	**0.039**	0.156	0.158	**0.039**
Example 2 (AR(1) 0.8 + AR(1) 0.05)						
$(50,50)$	0.142	0.142	**0.038**	0.145	0.147	**0.038**
$(50,100)$	0.148	0.148	**0.038**	0.150	0.152	**0.038**
$(100,100)$	0.147	0.148	**0.039**	0.151	0.153	**0.039**
$(100,200)$	0.150	0.150	**0.039**	0.153	0.155	**0.039**
$(200,200)$	0.152	0.152	**0.039**	0.155	0.158	**0.039**
Example 3 (AR(1) 0.8 + AR(1) 0.2))						
$(50,50)$	0.144	0.144	**0.038**	0.148	0.149	**0.038**
$(50,100)$	0.150	0.150	**0.038**	0.152	0.153	**0.038**
$(100,100)$	0.151	0.151	**0.039**	0.154	0.156	**0.039**
$(100,200)$	0.154	0.154	**0.039**	0.157	0.159	**0.039**
$(200,200)$	0.155	0.155	**0.039**	0.159	0.161	**0.039**

Finally in Table 2.5, we show the the 1 step and 5 steps ahead root mean square errors for the three examples. Overall, our method (SWPCA) has the smallest FRMSE for all the examples while the CPCA method performs the worst on the forecasting. The phenomenon tell us that the features extracted via the auto-covariance matrix is better than the ones from the covariance matrix, in the

Table 2.4: Covariance across Time and Sections of error terms

	Time dependence ($\widehat{\varepsilon}_{\cdot t}$)			Cross-sectional dependence ($\widehat{\varepsilon}_{p\cdot}$)		
(P,T)	CPCA	DPCA	SWPCA	CPCA	DPCA	SWPCA
Example 1 (AR(1) 0.8 + N(0,1))						
$(50,50)$	0.067	0.067	**0.005**	0.072	0.073	**0.005**
$(50,100)$	0.069	0.069	**0.004**	0.074	0.075	**0.003**
$(100,100)$	0.069	0.069	**0.003**	0.075	0.076	**0.003**
$(100,200)$	0.071	0.071	**0.003**	0.076	0.078	**0.002**
$(200,200)$	0.072	0.072	**0.002**	0.078	0.079	**0.002**
Example 2 (AR(1) 0.8 + AR(1) 0.05)						
$(50,50)$	0.067	0.067	**0.005**	0.072	0.073	**0.005**
$(50,100)$	0.070	0.070	**0.004**	0.075	0.076	**0.003**
$(100,100)$	0.069	0.069	**0.003**	0.075	0.076	**0.003**
$(100,200)$	0.071	0.071	**0.003**	0.076	0.078	**0.002**
$(200,200)$	0.072	0.072	**0.002**	0.078	0.079	**0.002**
Example 3 (AR(1) 0.8 + AR(1) 0.2))						
$(50,50)$	0.069	0.068	**0.005**	0.074	0.075	**0.005**
$(50,100)$	0.071	0.071	**0.004**	0.076	0.077	**0.003**
$(100,100)$	0.072	0.072	**0.003**	0.077	0.078	**0.003**
$(100,200)$	0.074	0.074	**0.003**	0.079	0.080	**0.002**
$(200,200)$	0.074	0.074	**0.002**	0.080	0.081	**0.002**

view of the forecasting accuracy. Moreover, the SWPCA has smaller forecasting error than the DPCA indicates that, extracting the different types of features sequentiality can benefit the forecasting more than mixing them together.

Table 2.5: 1 Step and 5 Steps Ahead Forecasting RMSE

	1 Step Ahead			5 Steps Ahead		
(P,T)	SWPCA	CPCA	DPCA	SWPCA	CPCA	DPCA
Example 1 (AR(1) 0.8 + N(0,1))						
$(50,50)$	**0.808**	0.829	0.822	**1.046**	1.053	1.051
$(50,100)$	**0.774**	0.793	0.787	**1.000**	1.004	1.004
$(100,100)$	**0.789**	0.812	0.804	**1.046**	1.054	1.053
$(100,200)$	**0.790**	0.814	0.807	**1.029**	1.035	1.034
$(200,200)$	**0.800**	0.819	0.812	**0.986**	0.996	0.993
Example 2 (AR(1) 0.8 + AR(1) 0.05)						
$(50,50)$	**0.827**	0.850	0.844	**1.039**	1.049	1.047
$(50,100)$	**0.802**	0.818	0.813	**1.041**	1.049	1.046
$(100,100)$	**0.804**	0.826	0.820	**1.025**	1.028	1.028
$(100,200)$	**0.790**	0.810	0.802	**0.993**	0.998	0.995
$(200,200)$	**0.787**	0.807	0.800	**0.986**	0.993	0.991
Example 3 (AR(1) 0.8 + AR(1) 0.2))						
$(50,50)$	**0.791**	0.809	0.805	**1.039**	1.045	1.043
$(50,100)$	**0.799**	0.812	0.808	**1.034**	1.037	1.035
$(100,100)$	**0.756**	0.771	0.766	**1.035**	1.039	1.040
$(100,200)$	**0.813**	0.825	0.822	**1.011**	1.018	1.015
$(200,200)$	**0.787**	0.803	0.799	**1.008**	1.017	1.015

In summary, our methods (SWPCA) provides a dimension reduction method that gives more accurate forecasts for the high dimensional time series data simulated in this section. In the Appendix A.1, more special simulated cases are presented.

2.6 Analysis of the US mortality data

In this section, we discuss the analysis of our method applied on age-specific mortality data of the US. The data is the mortality data of the US population in the Human Mortality Database (HMD) (91) obtained in July 2018. HMD

contains original calculations of death rates and life tables for the populations in 40 countries and areas, as well as the input data used in constructing those tables. The data we originally obtained from HMD includes the annual age-sex-specific information of the number of exposures to risk, the number of deaths, and the central death rate, for ages from 0 to 110+ (age 100 and above) during the period from 1933 to 2016. We focus our analysis on the age-specific central death rates of the total sex population. As the mortality data for advanced ages are measured sparsely which is mentioned in Lee and Carter [1992], death rates for the older age group (from age 91 to 110+) are summarized and incorporated into a modified death rate for age 90+. In view of this, the annual age-specific death rates under study consist of a matrix data with 84 yearly observations (1933-2016) for 91 ages (0-90). Following Lee and Carter [1992], we consider the log transformed central death rates $[\ln(m_{p,t})]_{P\times T}$, where $P = 91, T = 84$, for modeling purposes. By doing so, we can guarantee that the estimated and predicted central death rates are non-negative. We show the better reconstruction and forecasting performance of our proposed method compared to the static PCA and dynamic PCA in this section. Moreover, we explain that the two-step dimension reduction is necessary on the mortality data by examining the factor loadings of the features. At last, we illustrate that improving the accuracy of the predicted death rates is crucial with two applications.

2.6.1 Stationarity

As the log central death rates are not stationary time series, we modified the first step of the dimension reduction part in our method (SWPCA) to deal with the non-stationary issue, which is summarized in Algorithm 2.

The difference of Algorithm 2 and the first step in Algorithm 1 is, instead of obtaining $\widehat{\boldsymbol{b}}_i$ by the eigenvectors of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1)^\top$, we get it from $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)^\top$. Because $\boldsymbol{y}_t$ is not stationary, $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1)^\top$ is not a good estimator for the pop-

Algorithm 2: Modified First Step of the Dimension Reduction

Input: Data $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T] \in \mathbb{R}^{P \times T}$;
$\boldsymbol{y}_t = (\ln(m_{1,t}), \ln(m_{2,t}), \ldots, \ln(m_{P,t}))^\top$.

Output: $\widehat{k}_{it}^{(1)}$, which is described in the first step of our method (see 2.2.8).

Dimension Reduction Step 1:

1 Compute $\boldsymbol{d}_t = \boldsymbol{y}_t - \boldsymbol{y}_{t-1}$ and $\overline{\boldsymbol{d}} = T^{-1}\sum_{t=2}^{T-1}\boldsymbol{d}_t$, $t = 2, 3, \ldots, T$;

2 Compute

$$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1) = \frac{1}{T-1}\sum_{t=2}^{T-1}(\boldsymbol{d}_{t+1} - \overline{\boldsymbol{d}})(\boldsymbol{d}_t - \overline{\boldsymbol{d}})^\top$$

and $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)^\top$;

3 Compute $\widehat{\boldsymbol{b}}_i : P \times 1$ by the eigenvector corresponding to the i^{th} largest eigenvalue of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)^\top$,where $i = 1, 2, \ldots, \widehat{r}_1$. $\widehat{r}_1$ is chosen by the method introduced in Section 2.2;

4 Compute $\overline{\boldsymbol{y}} = T^{-1}\sum_{t=1}^{T}\boldsymbol{y}_t$ and $\widehat{k}_{it}^{(1)} = \widehat{\boldsymbol{b}}_i^\top(\boldsymbol{y}_t - \overline{\boldsymbol{y}})$.

ulation lag 1 auto-covariance of $\boldsymbol{y}_t$.

We now explain the reason for using $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)^\top$. From $\boldsymbol{d}_t = \boldsymbol{y}_t - \boldsymbol{y}_{t-1}$, we have $\boldsymbol{y}_t = \boldsymbol{y}_{t-1} + \boldsymbol{d}_t = \boldsymbol{y}_{t-2} + \boldsymbol{d}_{t-1} + \boldsymbol{d}_t = \cdots = \sum_{i=-\infty}^{t}\boldsymbol{d}_i$. With the coefficients $\boldsymbol{b}_i$, $\boldsymbol{d}_i$ can be expressed as $\boldsymbol{d}_i = \sum_{j=1}^{P}\boldsymbol{b}_j\varphi_{ij}$ where $\varphi_{ij} = \boldsymbol{b}_j^\top\boldsymbol{d}_i$, thus

$$\boldsymbol{y}_t = \sum_{i=-\infty}^{t}\boldsymbol{d}_i = \sum_{i=-\infty}^{t}\left(\sum_{j=1}^{P}\boldsymbol{b}_j\varphi_{ij}\right) = \sum_{j=1}^{P}\left(\boldsymbol{b}_j\sum_{i=-\infty}^{t}\varphi_{ij}\right) = \sum_{j=1}^{P}\boldsymbol{b}_j\psi_{tj},$$

where $\psi_{tj} = \sum_{i=-\infty}^{t}\varphi_{ij}$. Thus when performing the dimension reduction, the coefficients to form a low-dimensional representation of $\boldsymbol{u}_t$ should be the same as those of $\boldsymbol{y}_t$. If $\boldsymbol{d}_t$ is stationary, $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)^\top$ is a good estimator for $\boldsymbol{\Sigma}_{\boldsymbol{d}}(1)\boldsymbol{\Sigma}_{\boldsymbol{d}}(1)^\top$. Then eigenvectors of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}(1)^\top$ are better estimators of the factor loadings than those of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1)\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}(1)^\top$. We did stationary tests on the lag 1 differenced series of each age separately, and more than 72% of the ages have a stationary result under significant level 0.1. This might not be enough to say $\boldsymbol{u}_t$ is stationary, but for this dataset, it is better than the original log central death rates.

Due to the same reason, we make the same modification for the static PCA and the dynamic PCA methods when comparing using the US mortality data. Thus from now on, "CPCA" and "DPCA" refers to the static PCA and dynamic PCA which deal with the non-stationary issue, respectively. In addition, for comparison purpose, we also apply the static PCA method without considering the non-stationary issue, which is exactly the same as the method in Lee and Carter [1992] and we call it "Lee-Carter" in the following sections for convenience.

2.6.2 Revisit the structure of the US mortality data

In this section, we have a further discussion about the suitability of the SWPCA for the US mortality data. We examine the variance and time serial dependence of the central death rates of the US. Because we modified the first step of the SWPCA according to Section 2.6.1, instead of examining the original data, we check the first difference of log central death rates for each age. That is, for each age p, we compute the variance and lag 1 autocorrelation (representing the time serial dependence) of $d_{p,2}, d_{p,3}, \ldots, d_{p,T}$, where $d_{p,t} = \log(m_{p,t}) - \log(m_{p,t-1})$, $p = 0, 1, \ldots, 90+$, and $T = 84$. The results are shown in Figure 2.4.

In Figure 2.4, the top red plot shows the variances of $d_{p,\cdot}$ for age $p = 0, 1, \ldots, 90+$, and the bottom blue line shows the lag 1 autocorrelation of $d_{p,\cdot}$. From the plot, we see that the variances of ages from 5 to 13 are larger than those of ages from 25 to 40, while the lag 1 autocorrelations of ages from 5 to 13 are smaller than those of ages from 25 to 40. This is the same structure with the Example 5 in the simulation (in Appendix A.1). In addition, we have seen previously that the death rates of all ages have similar patterns, which indicates that information from part of the ages can be borrowed to help with the forecasting of other ages. Thus the first step of the proposed method would like to use information from the ages with powerful forecasting ability, such as ages 25 to 40, to help with the forecasting of other ages with weak correlations, ages 5 to 13

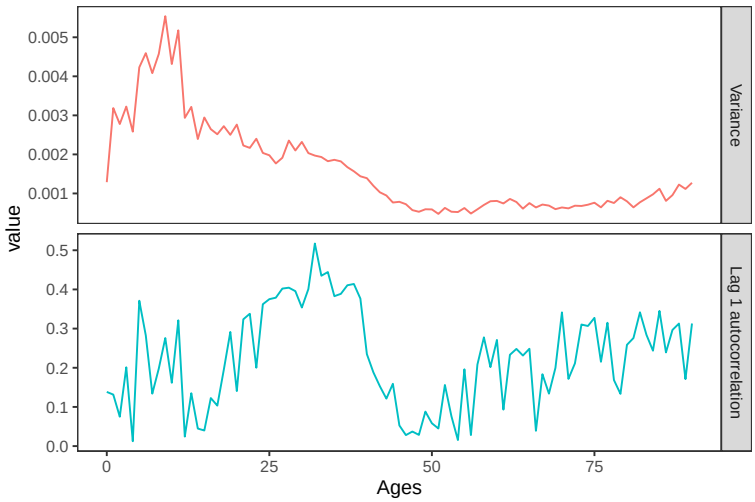


Figure 2.4: Variance and Time serial dependence of ages

for instance. On the other hand, the parts with powerful forecasting ability do not contain sufficient variations. For example, most of the variation is contained in younger ages while they do not all have large correlations. Therefore, the second step of our method utilizes static PCA to help retain sufficient variation of the original data, which is necessary for the final recovery for forecasting. As a result, SWPCA is particularly suitable for the US mortality data. In the next section, we examine the model fitting performance of the two-step dimension reduction method and illustrate the necessity of both steps using the estimated factor loadings in the following section.

2.6.3 Model fitting performance comparison

In this section, we check the performance of the SWPCA on reconstructing (fitting) the original data. We apply SWPCA, Lee-Carter, CPCA, and DPCA on the logarithm of central rates of death and compare the fitting performances using the root mean square error (RMSE). Based on the criteria in Section 2.2, all the methods choose only one feature. For SWPCA, we have $\widehat{r}_1 = 1$ and $\widehat{r}_2 = 1$. Table 2.6 and Table 2.7 show the RMSEs of the four methods for selected ages

(5, 25, 50, 65 and 85) and years (1933, 1953, 1993, and 2015) respectively, along with the overall RMSE of the whole data. From the tables, the RMSEs of the SWPCA are lower than those of the other three methods, which shows that SWPCA has a better fitting performance. As we described before, the two steps of the SWPCA guarantee that it captures sufficient variations of the original data and results in a good low-dimensional representation. In addition, we see that the RMSEs of Lee-Carter are smaller than those of CPCA. It implies that the fitting performance on the log central death rates is worse for static PCA if we revise the method to deal with non-stationarity. This phenomenon may be caused by special characteristics of the mortality data, which is interesting to explore further.

Table 2.6: RMSE, for some specific ages

Age	SWPCA	Lee-Carter	CPCA	DPCA
5	**0.051**	0.061	0.289	0.255
25	**0.058**	0.120	0.178	0.179
50	**0.051**	0.064	0.107	0.117
65	**0.038**	0.086	0.122	0.148
85	**0.045**	0.066	0.076	0.115
RMSE	**0.054**	0.080	0.146	0.149

Table 2.7: RMSE, for some specific years

Year	SWPCA	Lee-Carter	CPCA	DPCA
1933	**0.074**	0.148	0.179	0.162
1953	**0.047**	0.092	0.154	0.158
1993	**0.063**	0.082	0.147	0.150
2015	**0.076**	0.111	0.245	0.258
RMSE	**0.054**	0.080	0.146	0.149

We can also visualize the fitting performances of the four methods via plots. Figure 2.5 shows the actual and fitted log central rates of death for selected ages (5, 25, 50, 65 and 85) over all historical years from 1933 to 2016, while Figure 2.6 shows the actual and fitted log central rates of death for selected years (1933, 1953, 1993 and 2005) over all ages from 0 to 90+. The black lines represent the

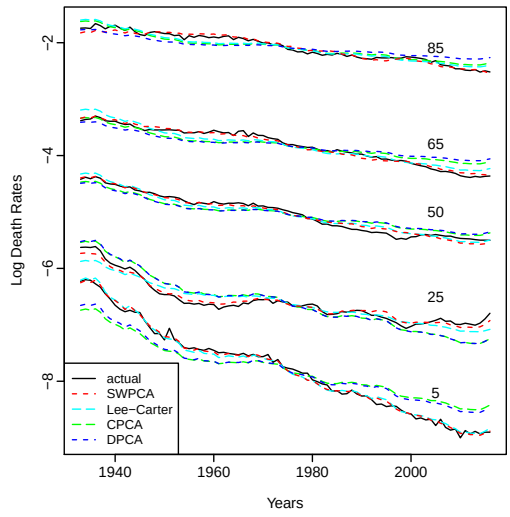


Figure 2.5: Log Death Rates, $1933 - 2016$ for ages 5, 25, 50, 65, 85. Actual and Fitted.

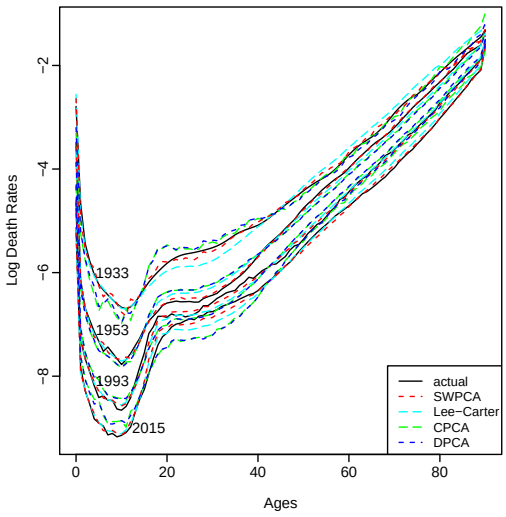


Figure 2.6: Log Death Rates, ages 0 to 90+ for years 1933, 1953, 1993, 2015. Actual and Fitted

actual log central rates of death; the red, light blue, green and blue dashed lines show the reconstructed log central rates of death using SWPCA, Lee-Carter, CPCA, and DPCA respectively. From Figure 2.5, we see that the SWPCA captures the time-serial patterns well for all selected ages even when there is curvature, such as the paths of ages 25, 65 and 85. However, the Lee-Carter, CPCA and DPCA failed to recover the time-serial dependence appropriately and hence provide worse reconstruction results than the SWPCA. From Figure 2.6, we see that the four methods provide similar reconstructions for ages 0 to 20. For ages 20 to 40, the mortality patterns changed and SWPCA shows a better reconstruction performance than the other three methods. The four models perform similarly again for ages 40 and above with the SWPCA's fitting performance slightly better than the other three, especially for the year 2015. Hence, it shows that the SWPCA captures both time-serial (time dimension) dependence and cross-sectional (age dimension) variation well and exhibits advantages over the other three methods especially when the mortality trends change.

2.6.4 Forecasting performance comparison

In this section, we show that our method can forecast the central death rates of the US more accurately than the other methods. We compare the forecasting performance of SWPCA, Lee-Carter, CPCA, DPCA, and the univariate ARIMA using rolling window out-of-sample forecasting. The univariate ARIMA model is fitted with time series of each age independently and the model structure is selected based on Bayesian information criterion (BIC); we refer to this as "individual" model. The individual model is included for comparison, as we would like to show that conducting dimension reduction before the forecasting is necessary, especially in the long term.

We use data of years from 2007 to 2016 as the test set and the historical data of previous years as the training set for modeling and testing purposes. Table 2.8 shows the forecasting root mean square errors (FRMSEs) of 1 to 25 steps ahead forecasts using the five methods. For each forecast, we have 10 rolling window sub-training sets for the 10 test years and the values presented in the table are the averages of the 10 rolling window FRMSEs. Figure 2.7 plots the results shown in Table 2.8. We see that as the length of prediction steps increases, the performance of all methods get worse. This is because the longer-term forecasting is always harder and contains more uncertainty. The individual model has the best forecasting accuracy when $h \leq 10$, while performs worse in the long-term compared with the SWPCA and the Lee-Carter. This is because the individual model focuses on capturing the mortality pattern of each age vector, which ignores the dependence among different ages and overlooks the cross-sectional common information. So, in the short term, individual factors dominate the forecasting performance, and the individual model performs best. However, the long-term mortality forecasting provides important assumptions for various actuarial practices and government policymaking, such as life insurance and annuities pricing and reserving, asset liability management of pension funds, and the solvency

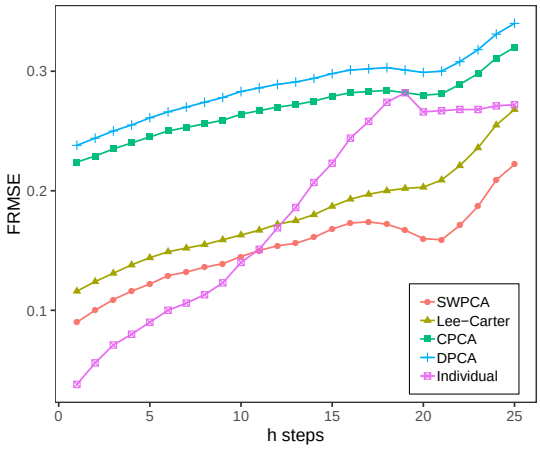


Figure 2.7: Comparison of the forecasting performance on the US data: Rolling Window FRMSE

analysis of social securities. In the long term, different ages share similar drivers of the mortality variation, such as technology innovation, health improvement, wars, and epidemics. So common factors dominate the forecasting performance in the long term, and dimension reduction plays a crucial role in recovering the common information from the high dimensional mortality data. Comparing the four dimension reduction methods (SWPCA, Lee-Carter, CPCA, and DPCA), we find that the SWPCA has the smallest FRMSEs for all h, and hence the best out-of-sample forecasting performance. The empirical analysis shows that SWPCA successfully extracts features with powerful forecasting ability and provides a good representation to recover the mortality forecasting from the features' forecasting.

2.6.5 Analysis of two-style factor model on mortality Data

Recall that the two-style factor model intends to capture two kinds of common features for mortality data among all ages: common temporal trends and common variation. Now we would like to analysis the necessity and the behavior of the proposed model on the mortality data under study.

Figure 2.8 provides the estimation of $\{u_{p,t} : t \geq 1\}$ for all $p = 1, \ldots, P$, which

Table 2.8: Comparison of the forecasting performance on the US data: Rolling Window FRMSE

h	SWPCA	Lee-Carter	CPCA	DPCA	Individual
1	0.090	0.116	0.224	0.238	**0.038**
2	0.100	0.124	0.229	0.244	**0.056**
3	0.109	0.131	0.235	0.250	**0.071**
4	0.116	0.138	0.240	0.255	**0.080**
5	0.122	0.144	0.245	0.261	**0.090**
6	0.129	0.149	0.250	0.266	**0.100**
7	0.132	0.152	0.253	0.270	**0.106**
8	0.136	0.155	0.256	0.274	**0.113**
9	0.139	0.159	0.259	0.278	**0.123**
10	0.145	0.163	0.264	0.283	**0.140**
11	**0.150**	0.167	0.267	0.286	0.151
12	**0.154**	0.172	0.270	0.289	0.169
13	**0.156**	0.175	0.272	0.291	0.186
14	**0.161**	0.180	0.275	0.294	0.207
15	**0.168**	0.187	0.279	0.298	0.223
16	**0.173**	0.193	0.282	0.301	0.244
17	**0.174**	0.197	0.283	0.302	0.258
18	**0.172**	0.200	0.284	0.303	0.274
19	**0.167**	0.202	0.282	0.301	0.282
20	**0.160**	0.203	0.280	0.299	0.266
21	**0.159**	0.209	0.281	0.300	0.267
22	**0.171**	0.221	0.289	0.308	0.268
23	**0.187**	0.236	0.298	0.318	0.268
24	**0.209**	0.255	0.311	0.331	0.271
25	**0.222**	0.268	0.320	0.340	0.272
Mean	**0.152**	0.180	0.269	0.287	0.181
Median	**0.156**	0.175	0.272	0.291	0.186

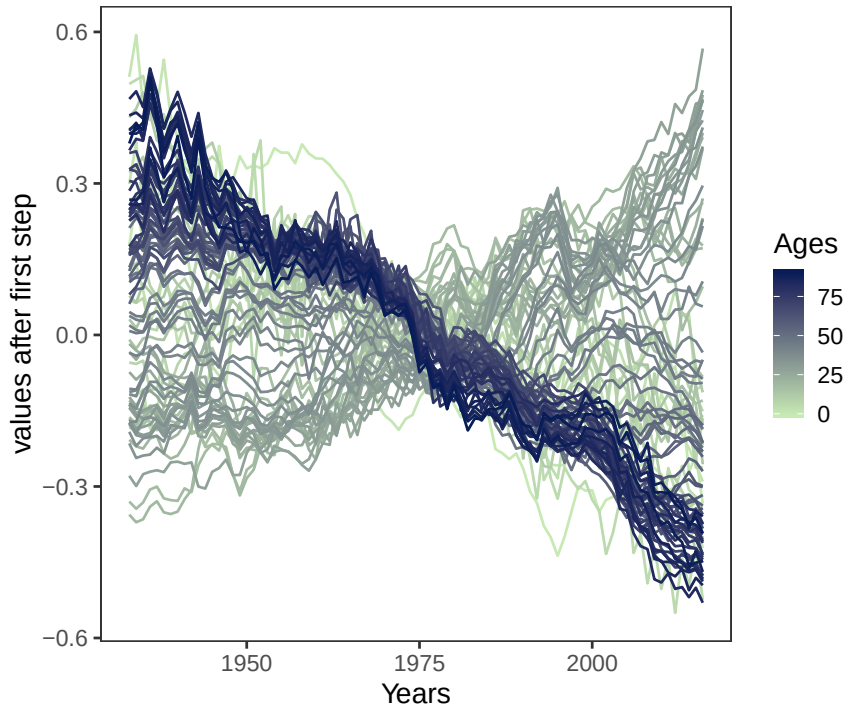


Figure 2.8: Estimation of u_t on the US mortality Data

is the residual of the first step. After extracting the common temporal trends in the first step, it is expected that there is relatively weak common temporal trend existed in the residual $u_{p,t}$. Compared with Figure 2.1 that illustrates the time-trends in original mortality data, Figure 2.8 indeed demonstrates weak common time-trends for all ages, in view of different time-tendency for the young ages from that of the old ages.

Next, we investigate the extracted features from the two kinds of factor models, respectively. As analyzed earlier, the estimation for the two parts is based on the eigendecomposition of the two matrices $\widehat{\boldsymbol{L}}_1$ and $\widehat{\boldsymbol{L}}_2$, respectively. The first line of Figure 2.9 shows all the eigenvalues of the two matrices. The spikeness is obvious and the ratio-based statistic will estimate r_1 and r_2 as 1 intuitively. Then the bottom line of Figure 2.9 provides the eigenvectors of $\widehat{\boldsymbol{L}}_1$ and $\widehat{\boldsymbol{L}}_2$ corresponding to their largest eigenvalue respectively. By comparing them, factor loadings from the two parts of factor models are quite different from each other.

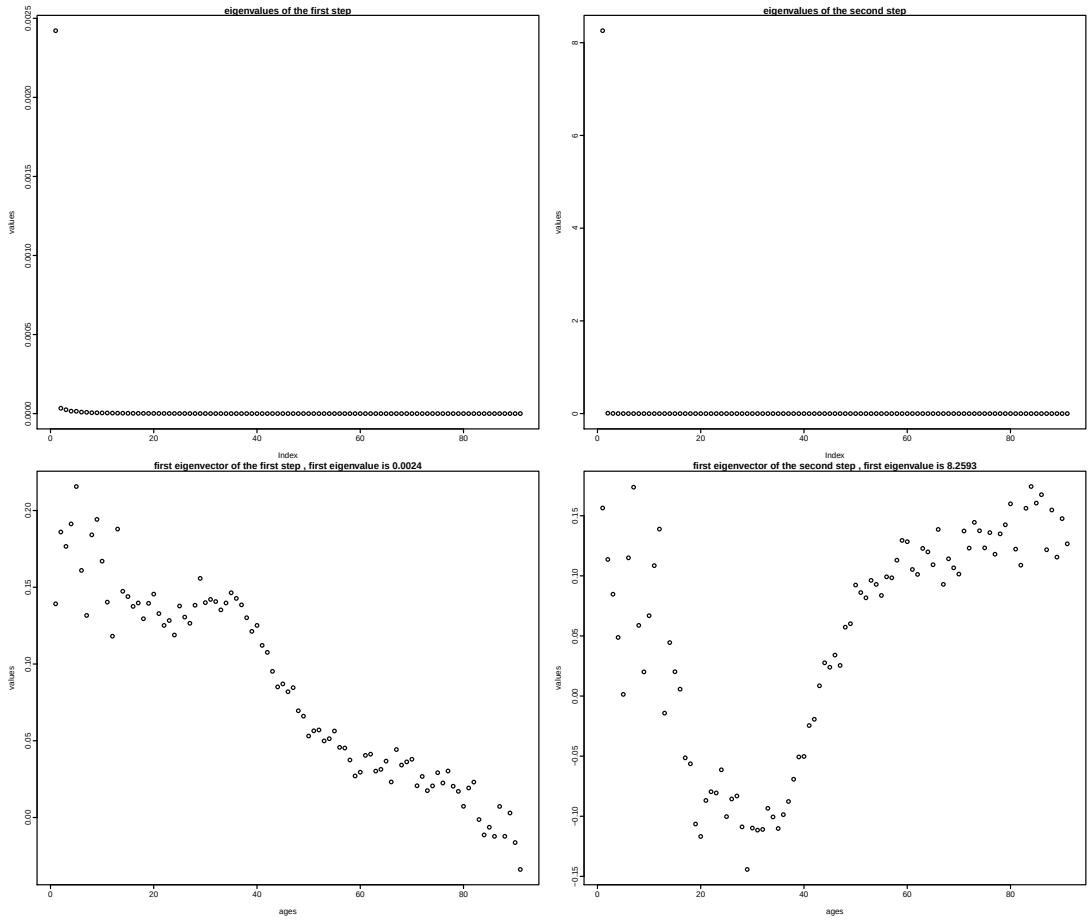


Figure 2.9: 1*st* Principal Component in Two Steps

A natural question may arise: other than conducting the second factor modelling, whether is it enough to keep two factors in the first step? Figure 2.10 shows that the second eigenvector of $\widehat{\boldsymbol{L}}_1$ is different from the first eigenvector extracted by the second step, which ensures the necessity of the second factor modelling. Roughly speaking, the second principal component (PC) in the first step represents weaker common temporal trend than the first PC, but stronger than the left PCs. However, the aim of our second step is to pursue features possessing most common variation of the residual after the first step. Although the extracted factor in the second step also has weaker common temporal trend

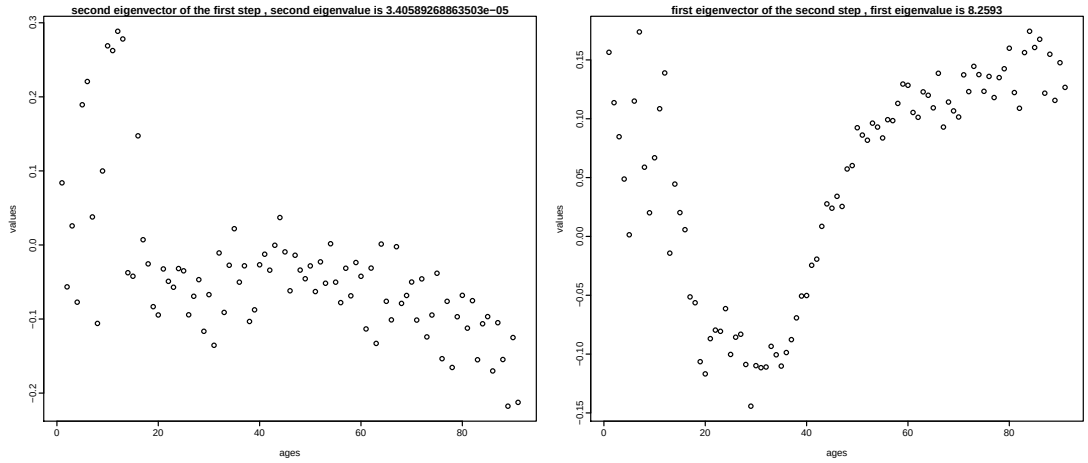


Figure 2.10: 2*nd* PC in Step 1 and 1*st* PC in Step 2

than that is extracted in the first step, it is not the second PC of the first step in view of Figure 2.10. Furthermore, in terms of eigenvalues of the matrix $\widehat{\boldsymbol{L}}_1$ in Figure 2.11, the second eigenvalue is not separable with others well except the first one. This phenomenon indicates that keeping the second PC may not increase sufficiently large amount of common temporal trends. In this case, the increased flexibility of keeping the 2nd PC will make this method undeserved.

2.6.6 Application of the mortality forecasting

In this section, we use the forecast of mortality to perform two applications: predicting the life expectancies and pricing the life annuities. The life expectancy describes the expected average remaining number of years prior to death for a person reached a specific age. Usually it can be reported in two different forms based on the mortality rates (period and cohort). The period life expectancy for a given year of each age is calculated based on the mortality rates for that single year, while the cohort life expectancy is estimated based on the mortality rates for the series of years in which the person will actually reach each succeeding age if the individual survives (The Board of Trustees of the Federal OASDI Trust Funds [2019]). For example, according to Table V.A4 and V.A5 in the 2019 report of

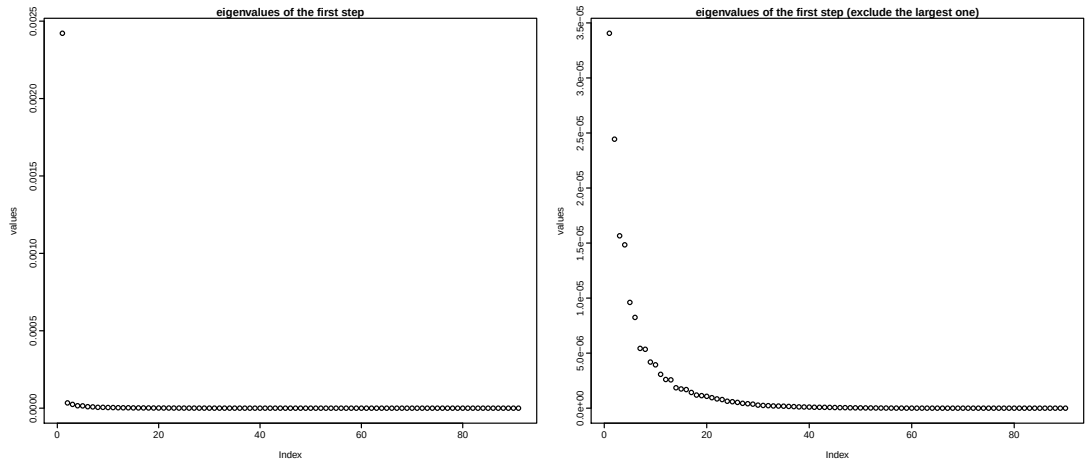


Figure 2.11: Eigenvalues of Step 1

The Board of Trustees of the Federal OASDI Trust Funds [2019], a male in the US aged 65 in year 2018 is expected to live another 18.1 years before death on a period basis while 18.9 years on a cohort basis. We will compare the estimated cohort and period life expectancy from our proposed method (SWPCA) with those from the Lee-Carter model. In addition, related to the cohort life expectancy, another interesting and crucial problem is, how much would an individual pay for an insurance which provides annual payments after the retirement until the death? We will compare the present values (price, per \$1) of the life annuities based on the estimated cohort life expectancies from different methods.

In the following part, we compute the actuarial life expectancy for an individual aged x at year T ($e_{x,T}$) as follows,

$$e_{x,T} = \sum_{t=1}^{w-x-1} {}_tp_{x,T},$$

where w is the assumed maximum age, and ${}_tp_{x,T} = \prod_{j=0}^{t-1}(1 - {}_1q_{x+j,T})$ is the probability that a person aged x at year T will survive to age $x+t$. For the period life expectancy, ${}_1q_{x+j,T} = m_{x+j,T}$, and for the cohort life expectancy, ${}_1q_{x+j,T} = m_{x+j,T+j}$, where $m_{x,t}$ is the death rate of a person aged x at year

t from the mortality table. In addition, for simplicity, we assume $1 - m_{90+,T}$ represents the probability that a person age 90 will survive to the maximum age w. Further, we calculate the present value of the life annuity ($PV_{x,T}$) for an individual purchased at age x in year T and beginning to make payments \$1 annually after age 66 until death or aged 90 (which one happens first) as below:

$$PV_{x,T} = \begin{cases} \sum_{t=1}^{90-x} {}_tp_{x,T}/(1+i)^t & \text{if } x \geq 66 \\ PV_{66,T+(66-x)}/(1+i)^{66-x} & \text{if } x < 66 \end{cases}$$

where $i = 2\%$, which is the interest rate, ${}_tp_{x,T} = \prod_{j=0}^{t-1}(1 - {}_1q_{x+j,T})$ and ${}_1q_{x+j,T}$ $= m_{x+j,T+j}$, which is on a cohort basis and the same with the calculation for the cohort life expectancy. We let the life annuities end at age 90 for simplicity as the mortality rates for extreme older ages need more detailed analysis, which is beyond the scope of this chapter. The age 66 is the retirement age for most individuals in the US. Hence, for an individual younger than 66, $PV_{x,T}$ is the price for a deferred life annuity. Similar calculation can be find in Cunningham et al. [2012] , McCarthy and Mitchell [2001], and Warshawsky [1988]).

In order to compare the out-of-sample performance of our method (SWPCA) and the Lee-Carter model, we define the data for years 1933 to 1986 as the training set and the data for the last 30 years (1987 to 2016) as the test set. We first forecast the mortality rates of the test set with the training set using the SWPCA and the Lee-Carter method, respectively. Then we calculate $e_{x,T}$ (cohort and period) and $PV_{x,T}$ using the actual mortality rates as well as the forecasting mortality rates from two methods, respectively.

With more accurate mortality forecasts, how much can the SWPCA method improve the prediction of the life expectancies and the pricing of the life annuities? Table 2.9 shows the forecast mean squared error (FMSE) and the forecast mean absolute error (FMAE) for the SWPCA method and the Lee-Carter model, which

are computed by,

$$\text{FMSE} = \frac{1}{N}\sum_x \sum_t (\hat{y}_{x,t} - y_{x,t})^2,$$
$$\text{FMAE} = \frac{1}{N}\sum_x \sum_t |\hat{y}_{x,t} - y_{x,t}|,$$

where $\hat{y}_{x,t}$ is the estimated value (computed with forecast death rates from the SWPCA or Lee-Carter), $y_{x,t}$ is the true value (computed with actual death rates), N is the number of estimates (it is different for the period and cohort life expectancies). It can be seen from the table that for all the three applications, the estimations from the SWPCA have smaller FMSEs and FMAEs comparing with those from Lee-Carter method. Particularly, from the FMAEs of the present values of life annuities, we can see that, on average, the pricing error is \$0.13 for Lee-Carter and only \$0.038 for SWPCA with annual payment \$1. The better performance of SWPCA is lead by the more accurately mortality forecasting.

Table 2.9: FMSE and FMAE of life expectancies (cohort and period) and present values of annuities (annual payment \$1 and interest rate 2%)

	FMSE			FMAE		
	period life expectancy	cohort life expectancy	pv of life annuity	period life expectancy	cohort life expectancy	pv of life annuity
Lee-Carter	0.593	0.076	0.027	0.687	0.211	0.130
SWPCA	**0.080**	**0.009**	**0.003**	**0.215**	**0.072**	**0.038**

Figure 2.12 and Figure 2.13 show the cohort and period life expectancies for an individual aged 66 at different years. The red line is the value computed from historical death rates, the green one is the value computed with the forecast from the SWPCA and the blue line is that from the Lee-Carter method. From Figure 2.12, we see that the three lines are close to each other before 1970, which is due to the less forecast involved in the calculation for those years. After 1970, when involving more forecast, the Lee-Carter method tends to estimate the life expectancies lower while the SWPCA is close to the true value with slightly

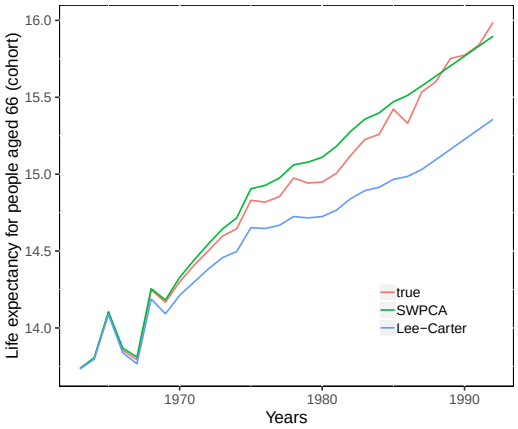


Figure 2.12: Comparison of the predicted life expectancies from the SWPCA and Lee-Carter with the true values (cohort)

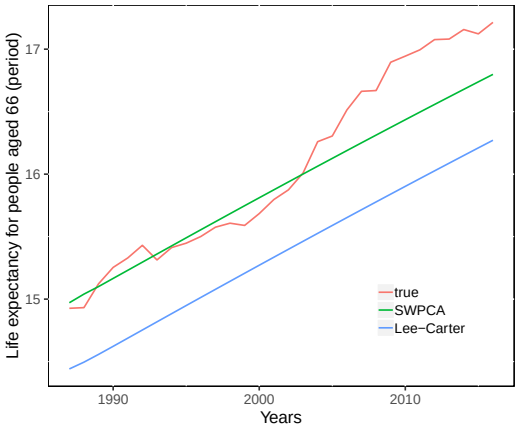


Figure 2.13: Comparison of the predicted life expectancies from the SWPCA and Lee-Carter with the true values (period)

higher estimations for some years. From Figure 2.13, we see that the output of SWPCA is always more close to the the true values while both the SWPCA and Lee-Carter tend to estimate lower for the second half the time horizon.

Table 2.10 exhibits the life expectancies (cohort and period) and the present values of annuities with annual payment \$1 and interest rate 2% for some selected ages and years (for some years and ages there are no forecast involves, hence we use $*$ to mark them). We can see that for the life expectancies, all the values from the SWPCA are closer to the true values than those from the Lee-Carter method, except that of $(2000, 75)$. On the other hand, the Lee-Carter method tends to price lower than the empirical true values for around \$0.20 to \$0.40 per \$1 of the life annuity, while SWPCA provides very accurately pricing with a maximum \$0.02 error per \$1 annual payment. Although the difference looks very small, it is indeed a big risk for life insurers or social security. To illustrate the financial impact on the industry, we can consider the pricing for individuals aged 65 in year 1990. The price from the Lee-Carter method is \$0.3 lower and from the SWPCA is \$0.02 higher per \$1 compared with the empirical true price. Suppose the annual payment for an individual is \$18000 and the number of people purchased

this insurance is 50000. Then according to the Lee-Carter method, the insurance company will have a \$270 million shortfall (270million $= 0.3 \times 18000 \times 50000$), which is a huge risk. On the other hand, SWPCA will have a \$18million surplus. Although it also mis-priced the insurance, the surplus will not put the company or the social security into a risky situation. In summary, our method improves the estimating of life expectancies and prices the life annuities more accurately by forecasting the mortality rates better.

Table 2.10: Selected life expectancies (cohort and period) and the present values of annuities (annual payment \$1 and interest rate 2%)

	period life expectancy			cohort life expectancy			pv of life annuity		
(year, age)	true	Lee-Carter	SW-PCA	true	Lee-Carter	SW-PCA	true	Lee-Carter	SW-PCA
(1950, 25)	45.81	*	*	50.14	49.62	50.11	5.72	5.55	5.72
(1960, 35)	37.59	*	*	40.86	40.34	40.84	6.98	6.77	6.97
(1970, 45)	29.19	*	*	31.99	31.45	31.97	8.50	8.25	8.50
(1980, 55)	22.59	*	*	23.87	23.29	23.84	10.37	10.06	10.36
(1990, 65)	15.95	15.29	15.85	16.52	15.95	16.51	12.64	12.27	12.63
(2000, 75)	9.60	9.45	9.89	10.09	9.68	10.12	8.63	8.33	8.65

2.7 Conclusion

This chapter focus on forecasting the US mortality data with a two-step dimension reduction method. Particularly, we analyzed the data structure of the age-specific central death rates of the US and proposed a new dimension reduction method especially suitable for forecasting this kind of data. We have found that the death rates for all the ages have similar patterns, which indicates common time-serial trend can be extracted to improve the forecasting of the data. In addition, variations among the death rates of all the ages is also crucial to provide more accurate fitting and benefit the forecasting. We make use of those characteristics to proposed the new method and find that this method can provide better forecasting results comparing with static PCA and dynamic PCA method.

To the best of our knowledge, this is the first work to especially consider the forecasting ability of dimension reduction. The novel dimension reduction method (SWPCA) can be seen as a two-style factor model, with estimations from the stepwise combination of static PCA (used in the Lee-Carter model) and dynamic PCA. It extracts two kinds of features that represent the common temporal trend and common variations receptively, which are both helpful for improving the forecasting accuracy. We simulated examples with the two-style factor model and we can clearly see that the SWPCA outperforms the other considered methods.

The detailed empirical analysis shows that the method is suitable and necessary for the mortality data in the US. Moreover, we find that the better forecasting of mortality from our method can improve the prediction of the corresponding life expectancy and life annuity. Hence the forecasting results of the SWPCA can be used to conduct important decisions in Actuarial science, such as providing advice for social security, pricing life insurances, and making the decision on required future cash reserves. Furthermore, we find in the long-term forecasting, recovering the mortality forecasting via features' forecasting is preferred than that via age-individually.

Chapter 3

Time-varying Factor Model for Mortality Forecasting

3.1 Introduction

Mortality forecasting is an important topic in various areas, such as demography, actuarial science and government policymaking. Most age-specific mortality data are high-dimensional time series. The factor model approach is one of the most popular methods to model high-dimensional time series, which represents the data matrix by a few latent common factors. Common factors describe common information shared by cross-sections, while factor loadings reflect the linear relationship between the original variables and the common factors. There is a large literature discussing factor models, including but not limited to Anderson [1963], Pena and Box [1987]), Stock and Watson [2002], Bai and Ng [2002], Bai [2009], Lam and Yao [2012] and Chang et al. [2018].

Many existing stochastic mortality models use the factor model approach. As an application of the classical factor model (with time-invariant factor loadings), Lee and Carter [1992] (Lee-Carter Model) is one of the most prominent methods for mortality forecasting, which is employed by the US Bureau of the Census as the benchmark model to predict long-run life expectancy (Hollmann et al. [1999]). The common factor extracted by the Lee-Carter model is defined as Mortality Index, and the factor loadings capture the relationship between the age

variables and the mortality index. Since there is only one factor in the Lee-Carter model, Booth et al. [2002], Renshaw and Haberman [2003] and Yang et al. [2010] extended the Lee-Carter framework to incorporate more common latent factors for mortality modeling in different countries. To deal with possible outliers in the mortality index, Li and Chan [2005] proposed an outlier-adjusted model by combining the Lee-Carter model with time series outlier analysis. Additionally, Booth et al. [2006] compared the Lee-Carter model with four other variants by applying them to mortality data of multiple populations. Tuljapurkar et al. [2000] examined mortality rates over five decades for the G7 countries using the Lee-Carter model. Lundström and Qvist [2004] and Booth et al. [2004] applied the Lee-Carter model to mortality data of Sweden and Australia, respectively. A summary of the variants of the Lee-Carter model is discussed in Booth and Tickle [2008].

In the existing literature of mortality factor models, factor loadings, which capture the relationship between age variables and latent common factors, are usually assumed to be time-invariant over time (we call factor models with time-invariant factor loadings 'classical factor models' in this chapter). For example, in Lee-Carter model, there is only one factor and the time-invariant factor loading represents the age-related sensitivity to the mortality improvement. However, since mortality datasets typically span a long period of time, it is restrictive to assume that the factor loadings are time-invariant. Driving forces such as medical improvement of certain diseases, environmental changes, and technological progress may influence the relationship of different variables significantly. Booth et al. [2002] studied the violation of the invariance assumption in the mortality data of Australia and suggested to find an optimal fitting period during which the factor loadings were invariant to improve the fit of the classical model. Their approach, however, needs to manually select the fitting period and hence loses the information of early years. In recent years, there is a rich literature on time-varying factor models to capture the dynamics and structural changes in factor

loadings for macroeconomic variables modeling, for example, see Breitung and Eickmeier [2011] and Chen et al. [2014]. However, there has been no literature on mortality modeling which allows factor loadings to change smoothly over time, to the best of our knowledge. Li and O'Hare [2017] and Li et al. [2015] used semi-parametric approaches to extend the CBD models [Cairns et al., 2009] by allowing for time-varying coefficients, which can free model assumptions and show superior short-term forecasting performance. However, CBD models are only suitable for old-age mortality modeling, and the factors (regressors) are observable. Unfortunately, for Lee-Carter model and many of its variants, the factors are unobserved, which makes it difficult to model and estimate. To fill those gaps, we introduce a factor model with time-varying factor loadings as an extension of the classical factor model based on Su and Wang [2017]. By developing corresponding estimation and forecasting methods, this new model can be used for mortality modeling and forecasting.

As the time-varying factor model allows for time-varying factor loadings, it provides more flexibility in model fitting, which, however, also poses challenges in model forecasting. Besides forecasting the common factors, factor loadings also need to be extrapolated into the future. In this chapter, we provide two forecasting methods of the factor loadings, one uses the local linear regression to roll over the time-varying factor loadings into the future; while the other one inherits the value of the factor loading from the last time period and remain invariant in the future. These two forecasting methods are called the local regression method and the naive method, respectively. Their details are described in Section 3.2. Empirical results using the mortality data from different populations show that the time-varying factor models provides more accurate out-of-sample forecasting results than the classical factor model.

The existing literature suggests that different forecasting horizons may favour different models. For example, Bell [1997] found that a simple random walk with drift model for age-specific mortality rates yields the most accurate 1-step-ahead

forecast compared to other six methods on the US data. Hyndman and Ullah [2007] introduced a method which outperformed the method proposed by Lee and Miller [2001] in the long-term forecasting. Specifically, we have found in the literature that semi-parametric or non-parametric methods may be more suitable for short-term forecasting. For example, the semi-parametric model developed in Li et al. [2015] can produce superior 5-year-ahead forecasting results. CMI [2009] employed the P-splines model (Currie et al. [2004]) for short-term forecasting to generate the initial rates of mortality improvement. Our empirical applications in Section 3.5.3 also suggest that the time-varying model based on local regression (non-parametric forecasting) is better for short-term forecasting, while the time-varying model based on naive method (parametric forecasting) is better for long-term forecasting. Then where is the optimal 'boundary' between short-term (based on the local regression method) and long-term (based on the naive method) forecasting? We propose a novel approach based on change point analysis [Bai, 2010] to estimate the optimal 'boundary' and apply it to mortality data of multiple countries. Additionally, we conduct simulation studies to show the performance of the time-varying factor model under different scenarios and investigate under which conditions it preforms better than the classical factor model.

The rest of the chapter is organized as follows. Section 3.2 introduces the time-varying factor model and its estimation approach. The forecasting methods based on the time-varying factor model are also discussed in detail. Section 3.3 discusses the relative advantages of the local regression method and the naive method in the short-term and long-term forecasting, respectively. We then propose an approach based on change point analysis to estimate the 'boundary' between short-term and long-term forecasting, which is favoured by the local regression method and the naive method respectively. Section 3.4 introduces the datasets and empirical evidence of time-varying factor loadings. Section 3.5 applies the proposed methods to age-specific mortality data of multiple countries and shows

the advantages of the proposed methods. Section 3.6 conducts simulation studies to investigate the performance of the time-varying factor model under different scenarios. Section 3.7 concludes the chapter. Appendix B.1 provides the gender-specific empirical results using the time-varying factor model. Appendix B.2 presents the estimations of the optimal boundaries for varies countries with a variety of forecasting horizons.

3.2 Time-varying factor model

Let $m_{x,t}$ denote the central death rate for age x in year t, where $x = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$. Thus, $\{m_{x,t}\}_{x=1,2,\ldots,N, t=1,2,\ldots,T}$ is a N-dimensional time series with T observations. Since mortality rates are always positive numbers, we use the log transformation to map the central death rates from $\mathbb{R}^+$ space to $\mathbb{R}$ space for modeling purposes. Assume a_x is the age-specific constants, which is the averages over time of the $\ln(m_{x,t})$. Then $\ln(m_{x,t}) - a_x$ can be modeled using the classical factor model, as follows:

$$\ln(m_{x,t}) = a_x + \boldsymbol{b}_x^\top \boldsymbol{k}_t + \varepsilon_{x,t}, \tag{3.2.1}$$

where $\boldsymbol{k}_t$ is a $R \times 1$ vector of common factors; $\boldsymbol{b}_x$ is a $R \times 1$ vector of factor loadings, capturing the impact of each common factor on age x (i.e. the age-related sensitivity to the mortality improvement); and $\varepsilon_{x,t}$ is the idiosyncratic error of $\ln(m_{x,t})$, which represents the components unexplained by the common factor. Here, $\boldsymbol{k}_t$, $\boldsymbol{b}_x$ and $\varepsilon_{x,t}$ are all unobservable components. Specifically, when $R = 1$, the classical factor model is equivalent to the Lee-Carter model. The single factor k_t was defined as the Mortality Index in the Lee-Carter model, and consequently the factor loading b_x represents the impact of the mortality index on the death rate of age x.

The classical factor model, however, is too restrictive when used to analyze

the mortality data. It assumes that, for each age, the factor loadings are time-invariant over time. Statisticians and economists have noticed that the relationship between many economic variables and common factors is not time-invariant. Our empirical analysis using mortality data in Section 3.5 also suggests time-varying factor loadings. Therefore, we develop a factor model to allow for factor loadings changing smoothly overtime.

We introduce the time-varying factor model based on the work of Su and Wang [2017], where factor loadings are modeled as nonrandom functions of time. Su and Wang [2017] provided a localized PCA method to consistently estimate the factors and time-varying factor loadings. Compared with Park et al. [2009], the time-varying factor model proposed by Su and Wang [2017] can capture more types of structural changes in factor loadings, including both continuous changes and abrupt structural breaks. Assume $\ln(m_{x,t}) - a_x$ follows the time-varying factor model with R unobservable common factors:

$$\ln(m_{x,t}) = a_x + \boldsymbol{b}_{x,t}^{\top}\boldsymbol{k}_t + \varepsilon_{x,t}, \tag{3.2.2}$$

where notations above are the same as the classical factor model, except for the factor loadings. Here, each component of the factor loading $\boldsymbol{b}_{x,t}$ is assumed to be a deterministic function of t/T: $\boldsymbol{b}_{x,t} = \boldsymbol{b}_x(t/T)$, where each component of $\boldsymbol{b}_x(\cdot)$ is an unknown piece-wise smooth function of t/T. The time-varying factor model can be seen as a generalization of the classical factor model. If $\boldsymbol{b}_x(\cdot)$ is time-invariant over time, which is a special case of the piece-wise smooth function, the time-varying factor model will degenerate to the classical factor model. Generally speaking, the assumption that factor loadings are time-invariant is too restrictive to hold in most settings. However, the time-varying factor model can relax this assumption by allowing for both continuous structural changes and abrupt changes in factor loadings, which can also benefit the mortality forecasting.

3.2.1 Identification problem

Similar to the classical factor model, there exists an identification problem in the time-varying factor model. Actually, at each time point t, and for any arbitrary $R \times R$ invertible matrix $\boldsymbol{H}_t$, we have $\boldsymbol{b}_{x,t}^\top \boldsymbol{k}_t = \left(\boldsymbol{H}_t^{-1}\boldsymbol{b}_{x,t}\right)^\top \left(\boldsymbol{H}_t^\top \boldsymbol{k}_t\right)$. Since an arbitrary $R \times R$ invertible matrix has R^2 free elements, R^2 restrictions are needed in parameter estimations so that $\boldsymbol{b}_{x,t}$ and $\boldsymbol{k}_t$ can be identified separately. Define $\boldsymbol{B}_t = (\boldsymbol{b}_{1,t}, \boldsymbol{b}_{2,t}, \ldots, \boldsymbol{b}_{N,t})^\top$ and $\boldsymbol{K} = (\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_T)^\top$. Then the two sets of restrictions to solve the issue of identification are as follows: $\boldsymbol{K}^\top \boldsymbol{K}/T = \mathbb{I}_R$ and $\boldsymbol{B}_t^\top \boldsymbol{B}_t$ = a diagonal matrix, where $\mathbb{I}_R$ is an $R \times R$ identity matrix. The first normalization condition imposes $R \times (R+1)/2$ restrictions on the parameters, and the remaining $R \times (R-1)/2$ restrictions are obtained by requiring the second constraint. These restrictions could uniquely determine the factors $\boldsymbol{K}$ and the factor loadings $\boldsymbol{B}_t$ (only up to a sign change, i.e., $-\boldsymbol{K}$ and $-\boldsymbol{B}_t$ also satisfy the two sets of restrictions). When $R = 1$, only one restriction is needed to identify parameters. We choose to use the same normalization condition as Lee and Carter [1992], that is, we normalize the $\boldsymbol{b}_{x,t}$ to sum to unity for each t. In this way, we can directly compare the results of our new method with that of the Lee-Carter model.

3.2.2 Estimation method

The estimation method for the time-varying factor model is proposed by Su and Wang [2017]. Let $r \in \{1, \ldots, T\}$ be a fixed year. Since we have assumed that each component of $\boldsymbol{b}_{x,t} : [0,1] \to \mathbb{R}$ is a piece-wise smooth function, we have:

$$\boldsymbol{b}_{x,t} = \boldsymbol{b}_x(\frac{t}{T}) \approx \boldsymbol{b}_x(\frac{r}{T}) = \boldsymbol{b}_{x,r}, \text{ when } \frac{t}{T} \approx \frac{r}{T}.$$

Thus, the mortality rate $\ln(m_{x,t})$ can be approximated by:

$$\ln(m_{x,t}) \approx a_x + \boldsymbol{b}_{x,r}^{\top}\boldsymbol{k}_t + \varepsilon_{x,t}, \text{ when } \frac{t}{T} \approx \frac{r}{T}.$$

In order to estimate the factors and time-varying factor loadings, we consider the following local weighted least squares problem:

$$\min_{\{\boldsymbol{b}_{x,r}\}_{x=1}^{N},\{\boldsymbol{k}_t\}_{t=1}^{T}} (NT)^{-1} \sum_{x=1}^{N}\sum_{t=1}^{T} \left(\ln(m_{x,t}) - a_x - \boldsymbol{b}_{x,r}^{\top}\boldsymbol{k}_t\right)^2 K_h\left(\frac{t-r}{T}\right), \quad (3.2.3)$$

subject to the identification constraints as discussed in Section 3.2.1. In the objective function in Equation (3.2.3), $K_h(x) = h^{-1}K(x/h)$, where $K(\cdot)$ is a kernel function and h is a smoothing parameter called "bandwidth". We will show that the optimization problem of Equation (3.2.3) can be solved using the same estimation method for the classical factor model.

We have known that the mortality rates can be approximated by $\ln(m_{x,t}) - a_x \approx \boldsymbol{b}_{x,r}^{\top}\boldsymbol{k}_t + \varepsilon_{x,t}$ when $\frac{t}{T} \approx \frac{r}{T}$. Multiplying both sides of the equation by

$$K_{h,tr}^{1/2} := \left(K_h\left(\frac{t-r}{T}\right)\right)^{1/2} = \left(\frac{1}{h}K\left(\frac{t-r}{Th}\right)\right)^{1/2},$$

we obtain a transformed model as:

$$K_{h,tr}^{1/2}(\ln(m_{x,t}) - a_x) \approx K_{h,tr}^{1/2}\boldsymbol{b}_{x,r}^{\top}\boldsymbol{k}_t + K_{h,tr}^{1/2}\varepsilon_{x,t}, \text{ when } \frac{t}{T} \approx \frac{r}{T}.$$

Then we can define matrices

$$\boldsymbol{M}^{(r)} = \left(\boldsymbol{M}_1^{(r)}, \ldots, \boldsymbol{M}_N^{(r)}\right),$$

$$\boldsymbol{\varepsilon}^{(r)} = \left(\boldsymbol{\varepsilon}_1^{(r)}, \ldots, \boldsymbol{\varepsilon}_N^{(r)}\right),$$

and

$$\boldsymbol{K}^{(r)} = \left(K_{h,1r}^{1/2}\boldsymbol{k}_1, \ldots, K_{h,Tr}^{1/2}\boldsymbol{k}_T\right)^\top,$$

where $\boldsymbol{M}_x^{(r)} = \left(K_{h,1r}^{1/2}(\ln(m_{x,1}) - a_x), \ldots, K_{h,Tr}^{1/2}(\ln(m_{x,T}) - a_x)\right)^\top$ and $\boldsymbol{\varepsilon}_x^{(r)} = \left(K_{h,1r}^{1/2}\varepsilon_{x,1}, \ldots, K_{h,Tr}^{1/2}\varepsilon_{x,T}\right)^\top$ with $x = 1, 2, \ldots, N$. Therefore, the transformed model can be written in matrix form as follows:

$$\boldsymbol{M}^{(r)} \approx \boldsymbol{K}^{(r)}\boldsymbol{B}_r^\top + \boldsymbol{\varepsilon}^{(r)},$$

and the optimization problem above can also be written in matrix notation as:

$$\begin{aligned} &\min_{\boldsymbol{K}^{(r)}, \boldsymbol{B}_r} \operatorname{Tr}\left(\left(\boldsymbol{M}^{(r)} - \boldsymbol{K}^{(r)}\boldsymbol{B}_r^\top\right)\left(\boldsymbol{M}^{(r)} - \boldsymbol{K}^{(r)}\boldsymbol{B}_r^\top\right)^\top\right) \\ &s.t\ \boldsymbol{K}^{(r)\top}\boldsymbol{K}^{(r)}/T = \mathbb{I}_R \text{ and } \boldsymbol{B}_r^\top\boldsymbol{B}_r = \text{a diagonal matrix.} \end{aligned}$$

Concentrating out $\boldsymbol{B}_r = \boldsymbol{M}^{(r)\top}\boldsymbol{K}^{(r)}\left(\boldsymbol{K}^{(r)\top}\boldsymbol{K}^{(r)}\right)^{-1}$ (which is $\boldsymbol{M}^{(r)\top}\boldsymbol{K}^{(r)}/T$ under the normalization $\boldsymbol{K}^{(r)\top}\boldsymbol{K}^{(r)}/T = \mathbb{I}_R$), the optimization problem is converted to minimizing the objective function:

$$\operatorname{Tr}\left(\boldsymbol{M}^{(r)\top}\boldsymbol{M}^{(r)}\right) - T^{-1}\operatorname{Tr}\left(\boldsymbol{K}^{(r)\top}\boldsymbol{M}^{(r)}\boldsymbol{M}^{(r)\top}\boldsymbol{K}^{(r)}\right).$$

Thus, the original local weighted least squares problem is equivalent to maximizing

$$\operatorname{Tr}\left(\boldsymbol{K}^{(r)\top}\boldsymbol{M}^{(r)}\boldsymbol{M}^{(r)\top}\boldsymbol{K}^{(r)}\right),$$

subject to the restriction $\boldsymbol{K}^{(r)\top}\boldsymbol{K}^{(r)}/T = \mathbb{I}_R$, which is equivalent to the optimization problem of the classical factor model.

Our objective is to obtain estimators of the factors and factor loadings. A two-stage estimation procedure is used to estimate those parameters. Let $\widehat{\boldsymbol{K}}^{(r)}$

denote the estimated factor matrix of $\boldsymbol{K}^{(r)}$, and $\widehat{\boldsymbol{B}}_r = \left(\widehat{\boldsymbol{b}}_{1,r}, \widehat{\boldsymbol{b}}_{2,r}, \ldots, \widehat{\boldsymbol{b}}_{N,r}\right)^\top$ denote the estimator of the time-varying factor loading matrix $\boldsymbol{B}_r$. Then, $\widehat{\boldsymbol{K}}^{(r)}$ is $\sqrt{T}$ times eigenvectors corresponding to the largest R eigenvalues of the $T \times T$ matrix $\boldsymbol{M}^{(r)}\boldsymbol{M}^{(r)\top}$, and $\widehat{\boldsymbol{B}}_r$ is $\boldsymbol{M}^{(r)\top}\widehat{\boldsymbol{K}}^{(r)}\left(\widehat{\boldsymbol{K}}^{(r)\top}\widehat{\boldsymbol{K}}^{(r)}\right)^{-1}$ (it is $\boldsymbol{M}^{(r)\top}\widehat{\boldsymbol{K}}^{(r)}/T$ under the condition $\boldsymbol{K}^{(r)\top}\boldsymbol{K}^{(r)}/T = \mathbb{I}_R$). Therefore, in the first step, we can acquire estimators $\widehat{\boldsymbol{B}}_r$ of the factor loadings for $r = 1, \ldots, T$.

Based on the estimator $\widehat{\boldsymbol{B}}_r$ of the factor loading matrix we get in the first stage, we consider another least squares problem in the second stage to obtain the estimator of the factor $\boldsymbol{k}_t$. The objective function we would like to minimize is as follows:

$$\sum_{x=1}^{N}\left(\ln(m_{x,t}) - a_x - \widehat{\boldsymbol{b}}_{x,t}^\top \boldsymbol{k}_t\right)^2 \text{ for } t = 1, \ldots, T.$$

Since we already have $\widehat{\boldsymbol{b}}_{x,t}$ from the first stage, the answer to this minimization problem is

$$\widehat{\boldsymbol{k}}_t = \left(\sum_{x=1}^{N}\widehat{\boldsymbol{b}}_{x,t}\widehat{\boldsymbol{b}}_{x,t}^\top\right)^{-1}\left(\sum_{x=1}^{N}\widehat{\boldsymbol{b}}_{x,t}\left(\ln(m_{x,t}) - a_x\right)\right) \text{ for } t = 1, \ldots, T.$$

Thus, using the two-stage estimation method, we can obtain consistent estimators for both the factors and time-varying factor loadings.

Next, we discuss some issues in the kernel estimation.

Remark 3.1. ***Boundary kernel.*** *Usually, there exists a boundary bias issue in the kernel estimation. Instead of using the ordinary kernel function, it is suggested that a boundary kernel should be used to help us obtain some uniform results. Let $\lfloor a \rfloor$ represent the greatest integer less than or equal to a, then the*

boundary kernel we choose to use is as follows:

$$K^*_{h,tr} = h^{-1}K^*_r\left(\frac{t-r}{Th}\right) = \begin{cases} \frac{h^{-1}K\left(\frac{t-r}{Th}\right)}{\int_{-\frac{r}{Th}}^{1} K(u)\mathrm{d}u} & r \in [1, \lfloor Th \rfloor] \\ h^{-1}K\left(\frac{t-r}{Th}\right) & r \in (\lfloor Th \rfloor, T - \lfloor Th \rfloor] \\ \frac{h^{-1}K\left(\frac{t-r}{Th}\right)}{\int_{-1}^{(1-r/T)/h} K(u)\mathrm{d}u} & r \in (T - \lfloor Th \rfloor, T] \end{cases}.$$

Remark 3.2. ***The choice of bandwidth***. *For the nonparametric local smoothing method, it is important to determine the bandwidth for the kernel estimation. There are two ways to choose the bandwidth. One is to use a data-driven method such as the cross-validation. The other one is to use Silvermans rule of thumb to set the bandwidth, which is much easier to compute. Su and Wang [2017] have shown that choices of the kernel function and the bandwidth have little impact on the performance of the information criteria. Thus, in the following empirical analysis, we decide to use the Epanechnikov kernel and its corresponding Silvermans rule of thumb bandwidth, which is* $h = (2.35/\sqrt{12})T^{-1/5}N^{-1/10}$.

Remark 3.3. ***Determination of the number of factors***. *There are mainly two methods to determine the number of factors* R. *The first one is to use a BIC-type information criterion proposed by Su and Wang [2017]. Under certain assumptions, the new information criterion can correctly choose the true value of* R. *However, those assumptions may not hold in real data. Additionally, it is not easy to implement the out-of-sample forecasting if the chosen value of* R *is too large.*

The second method is based on the fact that the original local weighted least squares problem can be transformed into an optimization problem of the classical factor model. Therefore, the cumulative sum of eigenvalues can help us identify the number of factors. Let c *denote a cut-off value between* 0 *and* 1, *and* λ_k *represents the* k^{th} *largest eigenvalue of the matrix* $\boldsymbol{M}^{(r)}\boldsymbol{M}^{(r)\top}$, *then we can choose the value of* R *as* $\min\{R : \left(\sum_{k=1}^{R}\lambda_k\right) / \left(\sum_{k=1}^{N}\lambda_k\right) \geq c\}$. *In the following analysis,*

we will set the cut-off value as $c = 0.9$ *and empirical analysis shows that only one factor is enough to capture most characteristics of the mortality data, which is consistent with the Lee and Carter [1992] model.*

3.2.3 Forecasting method

We now consider how to make out-of-sample forecasting using the time-varying factor model. Since the factor loadings change over time, not only should we make predictions on the common factors, but we should also extrapolate the factor loadings for each age. We describe in the following the forecasting method for a single factor model ($R = 1$), for the simplicity of notations. Assume that based on the historical data we have acquired the estimated common factor and factor loadings using the method mentioned in Section 3.2.2.

In order to forecast the common factor, firstly we fit the common factor with ARIMA model. Since Akaike Information Criterion (AIC) is asymptotically equivalent to the cross-validation when the maximum likelihood estimation is used to fit the model (Stone [1977]), we choose AIC as the model selection criteria to find the most appropriate ARIMA model. After that, we can use the chosen model to forecast and obtain prediction intervals for the latent factor (see more details in Chapter 5 & 9 of Brockwell et al. [1991]).

The factor loading $b_{x,t}$ is assumed to be an unknown piece-wise smooth function of time t. For the purpose of extrapolating the factor loading $b_{x,t}$ into the future, we will adopt two different methods to achieve the goal:

1. *The naive method.* We simply assume that in the forecasting horizon, $b_{x,t}(t > T)$ is set as $b_{x,T}$. This is essentially a parametric forecasting method and similar to that in the classical model but with a different estimated value. The naive method using constant factor loading has a simple structure and could provide more stable forecasts in the long term.

2. *The local regression method.* This method is based on a nonparametric

regression method – the local linear regression, to flexibly estimate the deterministic function $b_{x,t}$ (See more details in Fan and Gijbels [1996],Friedman et al. [2001]). Similar method has also been applied in Li and O'Hare [2017]; Li et al. [2015]. The local linear regression can easily extend the most recent trends, which is more suitable for short-term forecasting.

We briefly describe the local regression method in the rest of this section. The main idea of the local linear regression is to fit the linear regression using only the observations in the neighborhood of a target point t_0. This so-called localization is achieved by using a weight function $K_\lambda(t, t_0) = K\left((t - t_0)/\lambda\right)$, where K is a kernel function and the index λ indicates the width of the neighborhood. One of the commonly used kernel functions with compact support is Epanechnikov kernel, which is adopted in this chapter. For the Epanechnikov kernel, the window width parameter λ is the radius of the support region, which can be estimated using out-of-sample validation. The weight function assigns a weight to each time point t based on the corresponding distance from t_0 (i.e., $|t - t_0|$). In such a way, the resulting estimated function is a smooth function.

Specifically for the forecasting of the time-varying factor loading of each age x, the local linear regression solves a separate weighted least square problem at each target point $T + h$ $(h = 1, 2, \dots)$:

$$\min_{\alpha(T+h),\beta(T+h)} \sum_{t=1}^{T+h-1} K_\lambda(t, T+h) \left(b_{x,t} - \alpha(T+h) - \beta(T+h)t\right)^2.$$

Note that the notations $\alpha(T+h)$ and $\beta(T+h)$ indicate that these two parameters under study vary with the point $T + h$ in local linear method.

Let $\boldsymbol{b}_x = (b_{x,1}, b_{x,2}, \dots, b_{x,T+h-1})^\top$, $\boldsymbol{X} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 2 & \dots & T+h-1 \end{pmatrix}^\top$, and $\boldsymbol{W}(T+h)$ denote the $(T+h-1) \times (T+h-1)$ diagonal matrix with the t^{th} diagonal element $K_\lambda(t, T+h)$. Then by using the weighted least squares esti-

mation, we can obtain the estimators for $\alpha(T+h)$ and $\beta(T+h)$ as follows:

$$\left(\widehat{\alpha}(T+h), \widehat{\beta}(T+h)\right)^{\top} = \left(\boldsymbol{X}^{\top}\boldsymbol{W}(T+h)\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\top}\boldsymbol{W}(T+h)\boldsymbol{b}_x.$$

To ensure that $\boldsymbol{X}^{\top}\boldsymbol{W}(T+h)\boldsymbol{X}$ is nonsingular, the bandwidth parameter λ in the kernel function should be selected properly in practice (More details can be find in Fan and Gijbels [1996]). Therefore the forecast factor loading at point $T+h$ is

$$\widehat{b}_{x,T+h} = \begin{pmatrix} 1 & T+h \end{pmatrix} \begin{pmatrix} \widehat{\alpha}(T+h) \\ \widehat{\beta}(T+h) \end{pmatrix}.$$

Note that for $h > 1$, the forecasts $\widehat{b}_{x,T+1}, \ldots, \widehat{b}_{x,T+h-1}$ are evolved in $\boldsymbol{b}_x$ when estimating the factor loading at the time $T+h$. Following this method, we can estimate the factor loadings for each age as a smooth function of time t and then extrapolate the factor loadings into the future. Combining with the predicted common factors, we can make out-of-sample predictions of the central death rates using the time-varying factor model.

3.3 Optimal 'boundary' estimation

Under the framework of time-varying factor model, we assume the factor loading $b_{x,t}$ is a function of time t. In Section 3.2.3, we introduced two different methods to extrapolate factor loading. One is a naive method, which is more suitable for long-term forecasting; and the other is based on local linear regression, which is more suitable for short-term forecasting. Then can we estimate the 'boundary' between short-term and long-term forecasting that divides the forecasting horizon according to the predictive power of the local regression method and the naive method?

To solve this problem, we first propose a new forecasting method, which is a

hybrid of two previously introduced methods. Assume T_0 is the number of years used in fitting the model and k ($k = 0, 1, 2, \cdots$) is the optimal boundary between short-term and long-term forecasting, favoured by the time-varying models based on local regression and the naive method, respectively. We have the point forecast estimation of mortality rate $ln(m_{x,t})$ for any given x, t and k ($k \geq 1$) using the hybrid method as

$$ln(\widehat{m}_{x,t}) = \begin{cases} \widehat{a}_x + \widehat{b}_{x,t} \cdot \widehat{k}_t & T_0 + 1 \leq t \leq T_0 + k \\ \widehat{a}_x + \widehat{b}_{x,T_0+k} \cdot \widehat{k}_t & t \geq T_0 + k + 1 \end{cases}$$

If $T_0 + 1 \leq t \leq T_0 + k$, the forecast of $\ln(m_{x,t})$ at time t is $\widehat{a}_x + \widehat{b}_{x,t} \cdot \widehat{k}_t$, where $\widehat{b}_{x,t}$ is the extrapolated factor loading at time t based on the local regression method. When $t \geq T_0 + k + 1$, the forecast at time t is $\widehat{a}_x + \widehat{b}_{x,T_0+k} \cdot \widehat{k}_t$, where $\widehat{b}_{x,T_0+k}$ is time-invariant and obtained using the extrapolated factor loading at time $T_0 + k$ using the local regression method. For $k = 0$, the forecast at time t ($t > T_0$) is just $\widehat{a}_x + \widehat{b}_{x,T_0} \cdot \widehat{k}_t$ using the estimated factor loading at time T_0. In this case, the hybrid method degenerates to the naive method. In view of this, $T_0 + k$ is the time boundary between short-term and long-term forecasting, and between choosing the local regression and the naive method. Given the value of k, the hybrid method applies the local linear regression for the first k periods in the forecasting horizon and keeps the factor loadings ($\widehat{b}_{x,T_0+k}$) unchanged thereafter, which combines the local regression and naive methods. Additionally, the hybrid method guarantees a consistent and smooth transition from short-term to long-term forecasting.

As discussed in Section 3.1, different forecasting horizons may favour different models. Generally, long-term forecasting benefits more from the historical long-term trend and short-term forecasting relies on the recent trend [Booth et al., 2002]. Since the local linear regression can easily extend the most recent trends, it is more suitable for short-term forecasting. However, as time goes by, the

recent trends become less and less reliable, which is not suitable for long-term forecasting. On the other hand, the naive method using constant factor loading is more suitable for the long-term forecasting, as it has a simple structure and would provide more stable forecasts in the long term. Compared with the classical factor model, the naive method provides more accurate estimations not only for the factor loadings but also for the common factors, which helps generate more accurate long-term forecasts.

Based on the hybrid forecasting method, we propose an estimation method of the optimal 'boundary' inspired by Bai [2010]. Assume the entire dataset has T years and we consider the first T_0 years of data as the training set, and the remaining data with size $T - T_0$ as the validation set. Given the value of k, we first fit the time-varying factor model using the training set, and then apply the hybrid forecasting method to the validation set. We consider all possible lengths of short-term (long-term) forecasting horizons (i.e., k) and find out an optimal one using least squares estimation. We describe the estimation procedure as follows.

For given x and k such that $1 \leq k \leq T - T_0 - 1$, define $\widehat{y}_{x,t}(k) = \widehat{a}_x + \widehat{b}_{x,t} \cdot \widehat{k}_t$ as the predicted value of $\ln(m_{x,t})$ from the hybrid forecasting method based on the time-varying factor model. When $T_0 + 1 \leq t \leq T_0 + k$, $\widehat{b}_{x,t}$ is forecasted by the local regression method; And when $T_0 + k + 1 \leq t \leq T$, $\widehat{b}_{x,t} = \widehat{b}_{x,T_0+k}$, where $\widehat{b}_{x,T_0+k}$ is the predicted factor loading at time $T_0 + k$ obtained via the local regression method. Then we define the sum of squared residuals for age x as

$$\begin{aligned} S_{x,T}(k) &= \sum_{t=T_0+1}^{T} \left(\ln(m_{x,t}) - \widehat{y}_{x,t}(k)\right)^2 \\ &= \sum_{t=T_0+1}^{T_0+k} \left(\ln(m_{x,t}) - \widehat{a}_x - \widehat{b}_{x,t} \cdot \widehat{k}_t\right)^2 \\ &\quad + \sum_{t=T_0+k+1}^{T} \left(\ln(m_{x,t}) - \widehat{a}_x - \widehat{b}_{x,T_0+k} \cdot \widehat{k}_t\right)^2, \end{aligned}$$

where $k = 1, 2, \ldots, T - T_0 - 1$. Here k represents the length of the short-term forecasting horizon or the 'boundary' between short-term (based on the local regression method) and long-term (based on the naive method) forecasting. The local linear regression is used to make forecasts from $T_0 + 1$ to $T_0 + k$; while the naive method (i.e. assuming $\widehat{b}_{x,t}$ doesn't change over the remaining period) is used to make forecasts from $T_0 + k + 1$ to T. We define

$$S_{x,T}(0) = \sum_{t=T_0+1}^{T} \left(\ln(m_{x,t}) - \widehat{a}_x - \widehat{b}_{x,T_0} \cdot \widehat{k}_t\right)^2 \quad \text{for} \quad k = 0$$

and

$$S_{x,T}(T - T_0) = \sum_{t=T_0+1}^{T} \left(\ln(m_{x,t}) - \widehat{a}_x - \widehat{b}_{x,t} \cdot \widehat{k}_t\right)^2 \quad \text{for} \quad k = T - T_0.$$

In this way, $S_{x,T}(k)$ is defined for each $k = 0, 1, \ldots, T - T_0$. Thus, the total sum of squared residuals (SSR) across all ages is defined as

$$SSR(k) = \sum_{x=1}^{N} S_{x,T}(k).$$

Hence the least squares estimator of the optimal 'boundary' is

$$\widehat{k} = \underset{0 \le k \le T - T_0}{argmin}\ SSR(k).$$

The estimated optimal 'boundary' between the short-term (based on local linear regression) and the long-term (based on naive method) forecasting is the time $\widehat{k}$ that leads to the smallest SSR.

3.4 Data

The mortality data used in this chapter are extracted from the Human Mortality Database (HMD) (91). Six countries are selected for the empirical analysis in

Section 3.4 and Section 3.5. For each country, age-sex-specific death rates are available annually for the entire population. The selected countries are shown in Table 3.1 along with the corresponding available time horizons, which will be used for empirical analysis.

Table 3.1: Time horizon for different countries

Country	start year	end year	length
AUSTRALIA	1921	2018	98
CANADA	1921	2016	96
FRANCE	1816	2017	202
ITALY	1872	2017	146
JAPAN	1947	2018	72
USA	1933	2017	85

The mortality data are generally available from age 0 to age 110+ for each year. Since measures of mortality at very old ages are unreliable [Lee and Carter, 1992], we decide not to use mortality data of age 91 and over in the following analysis and end up with $N = 91$ ages.

In order to investigate whether the factor loadings are time-varying or time-invariant in the empirical data. We conduct an exploratory data analysis by applying the Lee-Carter Model on the US mortality data with rolling-window time frames. We first divide the entire dataset into 44 subsets (each with 40 yearly observations) with the first subset from year 1933 to year 1972, the second subset from year 1934 to year 1973, and so on. We then fit the Lee-Carter model on each of the subset and extract the factor loading b_x for each time frame. We plot the factor loadings of some selected ages in Figure 3.1. We can see that the factor loadings possess different dynamic patterns for different ages and they are not time-invariant.

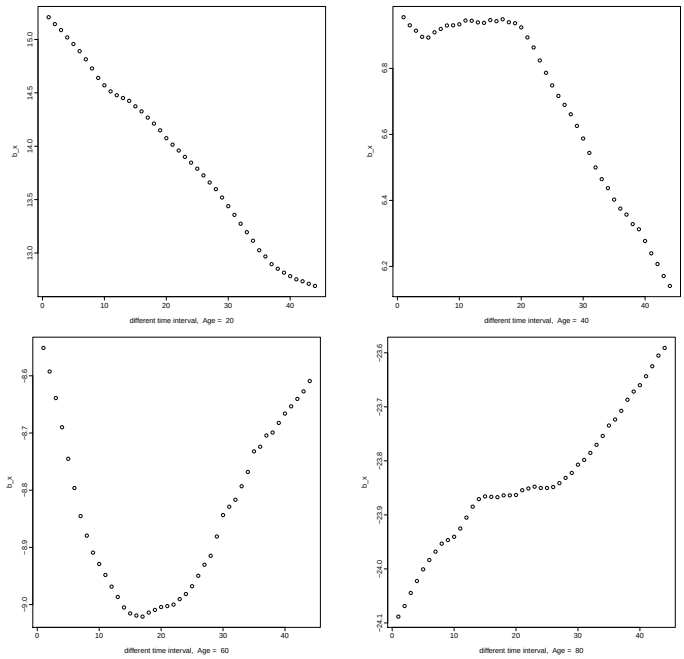


Figure 3.1: Factor loadings for ages 20, 40, 60, 80 over 44 rolling-window time frames

3.5 Empirical results and analysis

In the first two subsections, we present the application results of the time-varying factor model using age-specific mortality data of the US. We compare the time-varying factor models based on both naive and local regression forecasting methods with the classical factor model (i.e. Lee-Carter model) via out-of-sample forecasting performance. Empirical results by gender are provided in Appendix B.1. Section 3.5.3 further compares the forecasting performance across multiple countries and models based on different forecasting horizons. And in the Section 3.5.4, we estimate the optimal 'boundary' between short-term and long-term forecasting for different countries.

3.5.1 Model fitting

We fit the US mortality data using the estimation method of the time-varying factor model introduced in Section 3.2. The number of factors estimated is 1 ($\widehat{R} = 1$), which is consistent with the Lee-Carter model.

Using the model selection criteria AIC, we find that the common factor k_t, obtained from the time-varying factor model, follows an ARIMA $(1,1,0)$ with drift term. This model can capture most of the characteristics of the common factor. Our fitted model of the common factor k_t is as follows:

$$\underset{}{\bigtriangledown} k_t = \underset{(0.2791)}{-1.4116} + \underset{(0.1046)}{0.3271} \bigtriangledown k_{t-1} + e_t$$

where $\bigtriangledown$ refers to the first order differencing and e_t represents the error term. The numbers in the parentheses are the standard errors of the corresponding parameters. With the ARIMA model built above, we can then forecast the common factor into the future.

As a comparison, we also list the fitted ARIMA model of the common factor (The number of factors estimated is 1.) using the classical factor model[1] below:

$$\underset{}{\bigtriangledown} k_t = \underset{(0.2837)}{-1.4046} + \underset{(0.1051)}{0.3114} \bigtriangledown k_{t-1} + e_t$$

We can see that the ARIMA models of the common factors estimated from the time-varying factor model and the classical factor model are close. The estimated common factors, plotted in Figure 3.2, tend to decrease linearly and show similar dynamic patterns. The common factor is regarded as the index of the level of mortality, which capture major influence on death rates of all ages.

Figure 3.3 displays the comparison of the factor loadings between the time-varying factor model and the classical factor model for selected ages. Compared with time-invariant factor loadings (dashed lines), the time-varying factor loadings (the solid curves) change smoothly overtime, see Figure 3.3. It is interesting to notice that, no matter which age it is, the corresponding factor loadings always reach their own minimum or maximum values during 1960s or 1970s. For older people (over age 40), the factor loadings usually arrive at their maximum values

[1] In this case, the classical factor model has the same model structure as the Lee-Carter model.

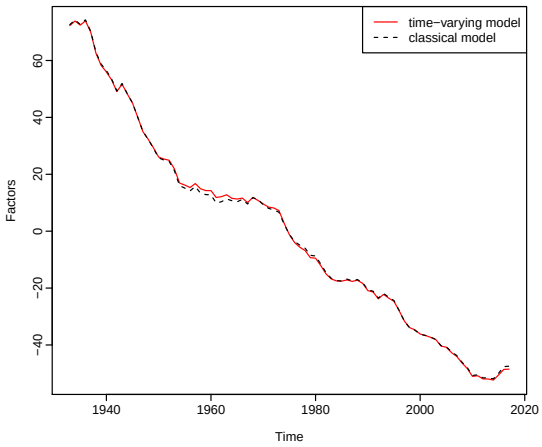


Figure 3.2: Plots of the estimated common factors for the time-varying factor model & the classical factor model

during 1960s or 1970s, which means the death rates of older people are more sensitive to the latent common factor during that period. For the younger ages (below age 40), however, the corresponding factor loadings reach their minimum values during the same period, which means the death rates of younger people are less sensitive to the latent factor during that time. The only exception is the factor loadings of the infant group, whose dynamic pattern is more similar to that of the older group.

Figure 3.4 shows the fitted death rates of both the time-varying factor model and the classical factor model with empirical observations for selected ages. Obviously, no matter which age it is, the time-varying factor model fits better than the classical factor model. We use the mean squared error (MSE)[2] to evaluate the goodness of fit. As a result, the overall MSE of the time-varying factor model is 0.001990, which is much smaller than that of the classical factor model, 0.006690 (three times bigger than the former one). Therefore, the time-varying factor

[2]The MSE for the time-varying model is computed as follows:

$$\text{MSE} = \frac{1}{NT} \sum_{x} \sum_{t} \left(\ln(m_{x,t}) - a_x - \widehat{\boldsymbol{b}}_{x,t}^{\top} \widehat{\boldsymbol{k}}_t \right)^2$$

Computation of the MSE for the classical model is the same as above except that $\widehat{\boldsymbol{b}}_{x,t}$ is replaced by $\widehat{\boldsymbol{b}}_x$.

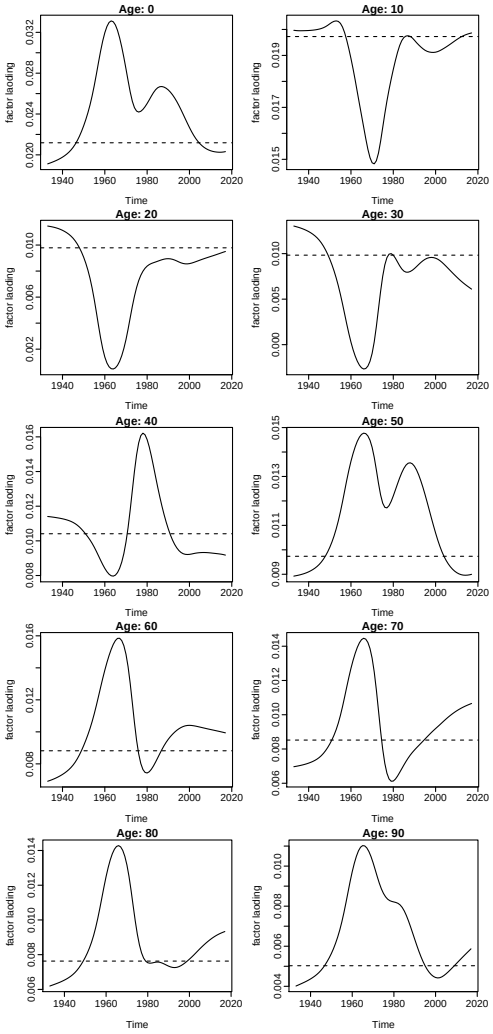


Figure 3.3: Plots of the estimated time-invariant factor loadings (dashed lines) & the time-varying factor loadings (solid lines) for age $0, 10, \ldots, 90$.

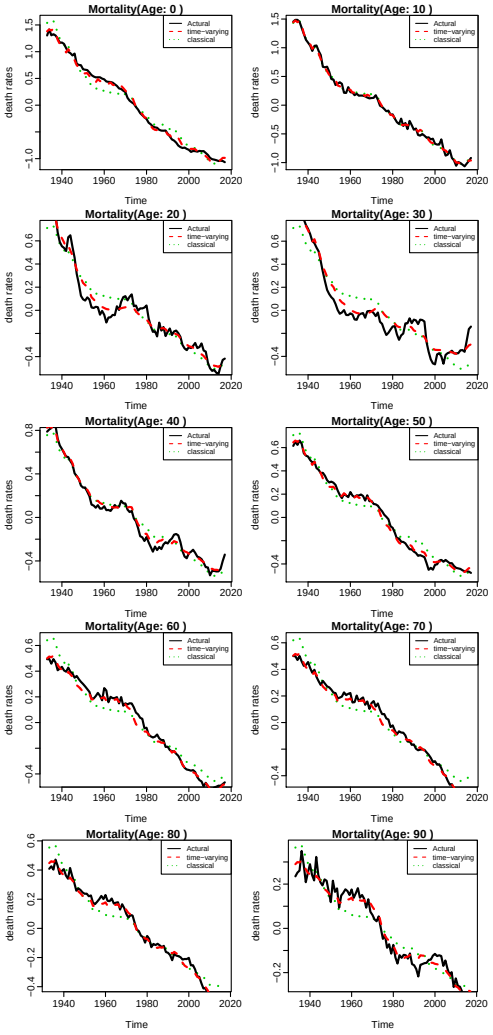


Figure 3.4: The actual data (black solid line) versus the fitted values from the time-varying model (red dashed line) and the classical model (green dotted line); the data have been log-transformed & demeaned.

model performs much better than the classical one with respect to the in-sample fitting.

Although the time-varying factor model works better in the fitting procedure, the problem of overfitting may exist due to the increased complexity of the model. Through the Monte Carlo simulation studies in Section 3.6, we will see that over-

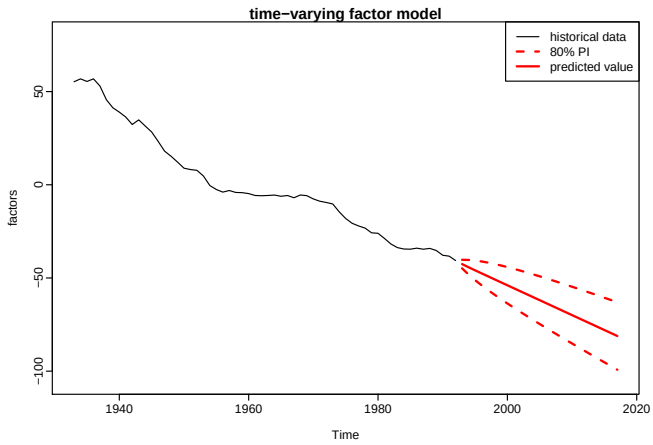


Figure 3.5: Out-of-sample forecast of the common factor, with the model fitted on 1933 to 1992 and the forecast horizon over 1993 to 2017; predicted value (red solid line), 80% PI (red dashed line)

fitting is harmful to forecasting. Usually, an overfitting model performs too well in the fitting sample to have good generalization ability in forecasting. Generally speaking, we can always improve a models in-sample fitting performance by increasing the complexity of the model, which, however, cannot guarantee a better forecasting performance in the future. Thus we will use the out-of-sample validation method to investigate whether the time-varying model can enhance the out-of-sample forecasting performance in the next subsection.

3.5.2 Out-of-sample forecasting

In this subsection, we use the original US mortality data over the first 60 years as the training set (from 1933 to 1992) to fit the models, and then forecast the mortality rates in the testing set (from 1993 to 2017) using the fitted models. The predicted values are compared with the actual data in the testing set to see which model is better at the out-of-sample forecasting. We apply the mean squared prediction error (MSPE) as the measure to evaluate the out-of-sample forecasting performance.

Figure 3.5 plots the historical and predicted values of the common factor of

the time varying factor model along with the associated 80% prediction intervals, which is based on the ARIMA model fitted in Section 3.5.1. The red solid downward line shows that the latent factor will keep declining in the future, and there is an 80% chance that a future observation will be covered by the corresponding prediction interval (represented by area between the red dashed lines).

Since the time-varying factor model and the classical factor model have similar common factors and the corresponding fitted ARIMA models, the forecasts of the common factors are close to each other too. Hence the major difference of prediction accuracy between the time-varying factor model and the classical factor model lies in the factor loadings. We extrapolate the factor loadings obtained from the time-varying factor model using both the naive method and local linear regression introduced in Section 3.2.3, respectively. We then forecast the mortality rates into the future using both the time-varying and classical factor models.

Figure 3.6 plots the estimated and extrapolated factor loadings of the time varying factor models. Figure 3.7 plots the actual data and predicted values using the three above-mentioned methods. From Figure 3.6, we see that the local regression method follows the recent historical trend of factor loadings, while the naive method stays at a constant level. Theoretically, if future factor loadings do not deviate significantly from the recent historical trend, the local linear regression may perform better than the naive method in forecasting. However, it may only be reasonable to assume that factor loadings will follow the local trends in the short-term. For long-term forecasting, this assumption is less reliable and the non-parametric forecasting method would lead to inferior results. In Section 3.5.3, we observe similar results in other countries' mortality forecasting. Hence, the naive method, with time-invariant forecasted factor loadings, is more suitable for long-term forecasting. And it may also be suitable for short-term forecasting if the long-term trend is consistent with the short-term trend. Since the benefit of using the local regression method decreases as the forecasting horizon increases,

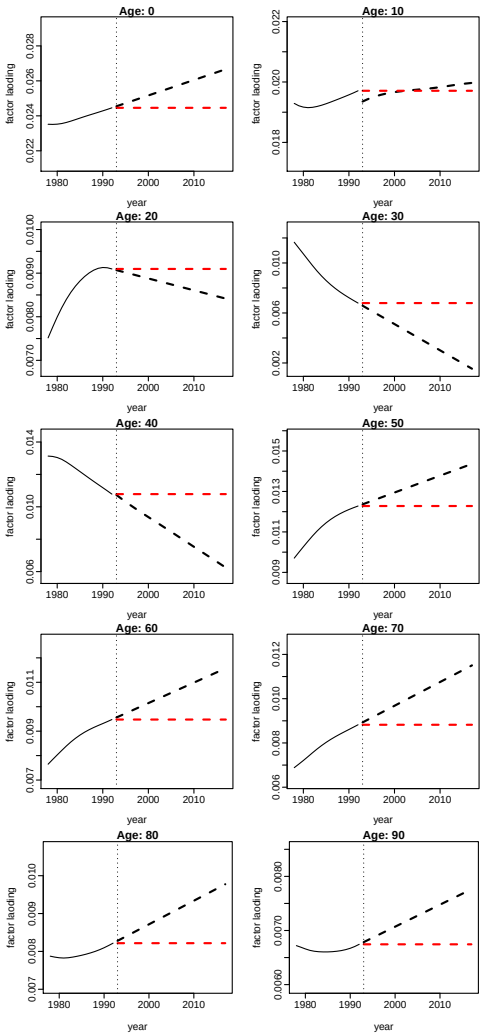


Figure 3.6: Plots of the estimated and extrapolated factor loadings based on naive method (red dashed lines) & local regression method (black dashed lines) for age $0, 10, \ldots, 90$.

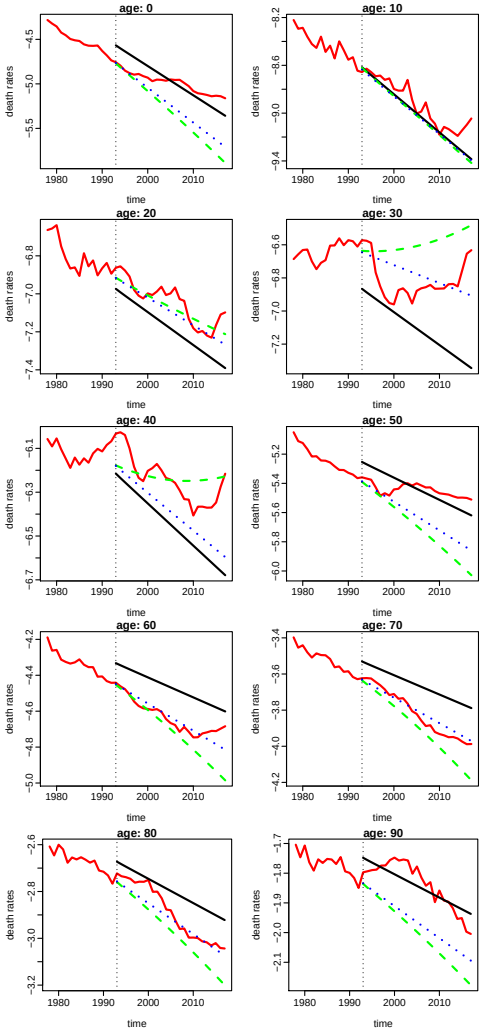


Figure 3.7: The actual data (red solid line) versus the predicted values from the time-varying model (naive method: blue dotted line; local linear regression: green dashed line) and the classical model (black solid line); the data have been log-transformed.

it is worthwhile to ask whether an optimal forecasting horizon exists for using the local regression method. This question will be answered in Section 3.5.4.

The empirical analysis suggests that, the time-varying factor model (based on the naive method) performs better than the classical factor model over the

entire forecasting horizon (1993-2017). Using the mean squared prediction error (MSPE) to evaluate the out-of-sample forecasting performance, we see that the overall MSPE for the classical factor model is 0.03085, while the overall MSPE for the time-varying factor model (based on the naive method) is only 0.01804. However, if we choose to use local linear regression to extrapolate factor loadings, the time-varying factor model performs worse than classical factor model, with MSPE of 0.04768.

Figure 3.8 shows the year-specific MSPE for the time-varying factor models and the classical factor model over the forecasting horizon 1993 to 2017. The year-specific MSPE is computed by averaging MSPE over all ages for each forecasting year. From Figure 3.8, we can see that for the majority years, the MSPE of the time-varying model with the naive forecasting method is always the smallest one. From Section 3.5.4, we can see the reason is that the optimal 'boundary' between short-term and long-term forecasting in this case is estimated to be 0. So the the time-varying model with the naive forecasting method is the best for both short-term and long-term forecasting. And, the MSPE for all the three methods are generally increasing over the years, as it is harder to forecast the farther future. We also notice that the time-varying factor model based on local regression method works better than the classical factor model over 1993 to 1995. However, it has the worst performance for longer-term forecasting. The time-varying model based on local regression method assumes the factor loadings change over time in the future, but it can only extend the recent trend, which may not be suitable for long-term forecasting. On the other hand, the classical factor model and the time-vary model based on naive method extrapolate factor loadings into future as constants, which is usually more suitable for long-term forecasting.

Next, we investigate the forecasting performance of the time-varying factor model at different ages. Figure 3.9 shows the age-specific MSPE for the time-varying factor models and the classical factor model. The yellow and pink plots

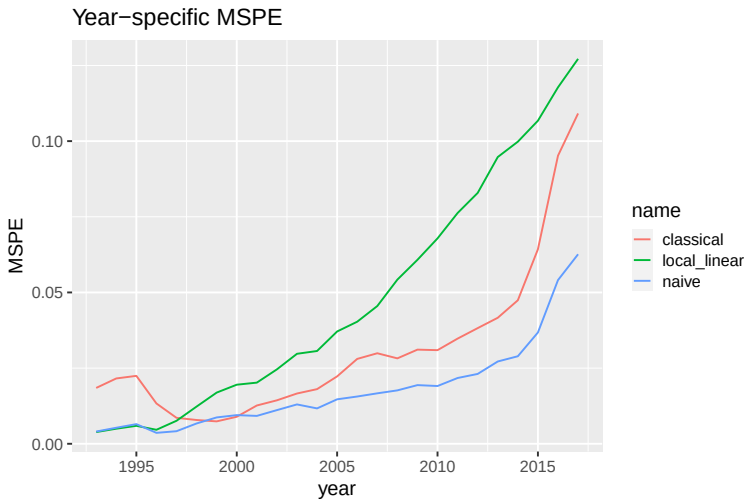


Figure 3.8: US: Year-specific MSPE for the time-varying model and the classical model over 1993 to 2017; for time-varying model, both the naive method and the local regression method are used

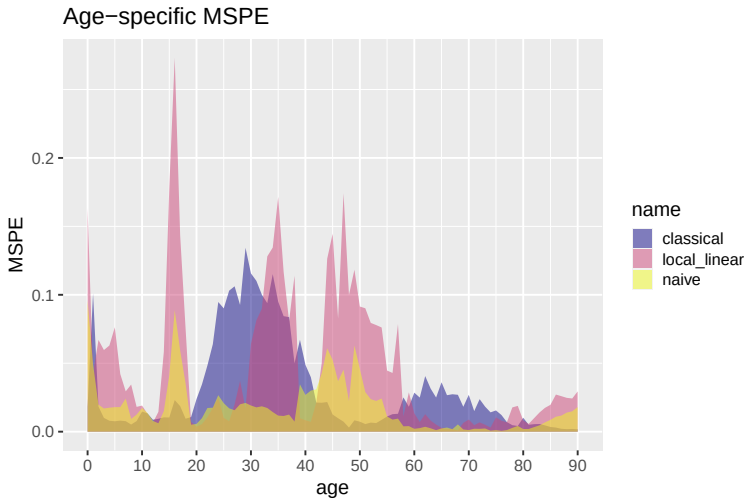


Figure 3.9: US: Age-specific MSPE for both the time-varying model and the classical model; for time-varying model, both naive method and local regression method are used

represent the MSPE of the time-varying factor model with naive method and local regression method for each age, respectively, and the purple plot represents the MSPE of the classical factor model. We find that the naive forecasting method based on the time-varying factor model is almost always better than local regression method for any age in this data. And roughly speaking, no matter

which extrapolation method we choose to use, the time-varying models provide more accurate forecasts than the classical factor model for age groups $20 \sim 40$ and $60 \sim 80$. However, for the age group $40 \sim 60$, the forecasting performance of the time-varying factor models is worse than that of the classical model. Thus, even though, by using naive extrapolation method, the time-varying factor model improves the overall performance (over 40% in terms of the MSPE) significantly, it cannot outperform the classical factor model for some ages. The main advantage of the time-varying model is to forecast mortality rates for the young adulthood ($20 \sim 40$) and the older adulthood ($60 \sim 80$).

3.5.3 Model comparisons for multiple countries

We apply and compare different models using mortality data of multiple countries. In particular, the functional data model proposed by Hyndman and Ullah [2007] is also considered for comparison purposes. It is a multi-factor extension of the Lee-Carter model, allowing for multiple age-time interaction terms to capture the complex structure of the data. Similar to the Lee-Carter model (i.e. the classical factor model), it is also commonly considered as a benchmark for mortality forecasting.

In Table 3.2, we present the results of the overall MSPE for different countries, forecast horizons and methods. We use the longest available dataset for training purposes, with different forecasting horizons listed in Table 3.2. Please refer to Section 3.4 for a brief description for mortality data of those countries. To investigate the performance of the short-term and long-term forecasting, we consider multiple forecasting horizons with different lengths, including 5, 10, 15, 20 and 25 years.

Table 3.2: Overall MSPE by country, forecast horizon and method (**functional**: functional data model, **classical**: classical factor model, **TV-Local Regression**: time-varying factor model based on local linear regression, **TV-Naive**: time-varying factor model based on naive method)

Country	Forecast Horizon	Functional	Classical	TV-Local Regression	TV-Naive
Australia	2014 ∼ 2018	**0.012**	0.037	0.013	0.014
	2009 ∼ 2018	0.022	0.049	**0.016**	0.019
	2004 ∼ 2018	0.036	0.064	**0.026**	0.028
	1999 ∼ 2018	0.082	0.094	0.045	**0.042**
	1994 ∼ 2018	0.053	0.112	0.089	**0.043**
Canada	2012 ∼ 2016	**0.012**	0.044	0.015	0.016
	2007 ∼ 2016	0.018	0.045	**0.016**	0.018
	2002 ∼ 2016	0.021	0.049	**0.019**	0.020
	1997 ∼ 2016	0.021	0.051	0.025	**0.020**
	1992 ∼ 2016	0.032	0.061	0.038	**0.024**
France	2013 ∼ 2017	**0.009**	0.047	0.0146	0.0154
	2008 ∼ 2017	**0.016**	0.054	0.021	0.022
	2003 ∼ 2017	**0.042**	0.081	0.049	0.047
	1998 ∼ 2017	**0.059**	0.101	0.068	0.061
	1993 ∼ 2017	**0.079**	0.128	0.096	0.085
Italy	2013 ∼ 2017	**0.012**	0.044	0.014	0.015
	2008 ∼ 2017	**0.018**	0.057	0.028	0.029
	2003 ∼ 2017	**0.047**	0.084	0.064	0.052
	1998 ∼ 2017	0.090	0.098	0.077	**0.058**
	1993 ∼ 2017	0.099	0.108	0.073	**0.067**
Japan	2014 ∼ 2018	0.012	0.031	**0.009**	**0.009**

Continued on next page

Table 3.2 – continued from previous page

Country	Forecast Horizon	Functional	Classical	TV-Local Regression	TV-Naive
	2009 ∼ 2018	**0.010**	0.034	0.017	0.011
	2004 ∼ 2018	0.039	0.072	0.023	**0.016**
	1999 ∼ 2018	0.024	0.055	0.018	**0.017**
	1994 ∼ 2018	**0.030**	0.072	0.037	0.034
USA	2013 ∼ 2017	**0.011**	0.028	0.013	0.014
	2008 ∼ 2017	**0.010**	0.024	0.015	0.014
	2003 ∼ 2017	0.017	0.025	0.024	**0.016**
	1998 ∼ 2017	0.111	0.029	0.031	**0.021**
	1993 ∼ 2017	0.026	0.031	0.048	**0.018**

From Table 3.2, we see that the time-varying models can significantly improve the out-of-sample forecasting performance compared with the classical factor model. The time-varying factor models and functional data model performs the best in most cases. The functional data model is especially suitable for mortality forecasting of the France data, as shown in Hyndman and Ullah [2007]. In addition, for the time-varying factor models, the local regression method tends to perform better for short-term forecasting, while the naive method is better for long-term forecasting. Compared with all the other models, the time-varying model based on the naive method show superior results for long-term forecasting, while the classical factor model always has the worst forecasting performance.

In Figure 3.10, by fixing the length of forecast horizon to be 25 years, we plot the year-specific MSPE for all the countries using different forecasting methods. Roughly speaking, the time-varying model based on the naive method always has the most accurate forecasts, while the classical factor model always performs the worst. Additionally, the time-varying model based on the local regression method usually performs well (similar to or better than the naive method) at the first few years, while it deteriorates as time goes by. For mortality data of

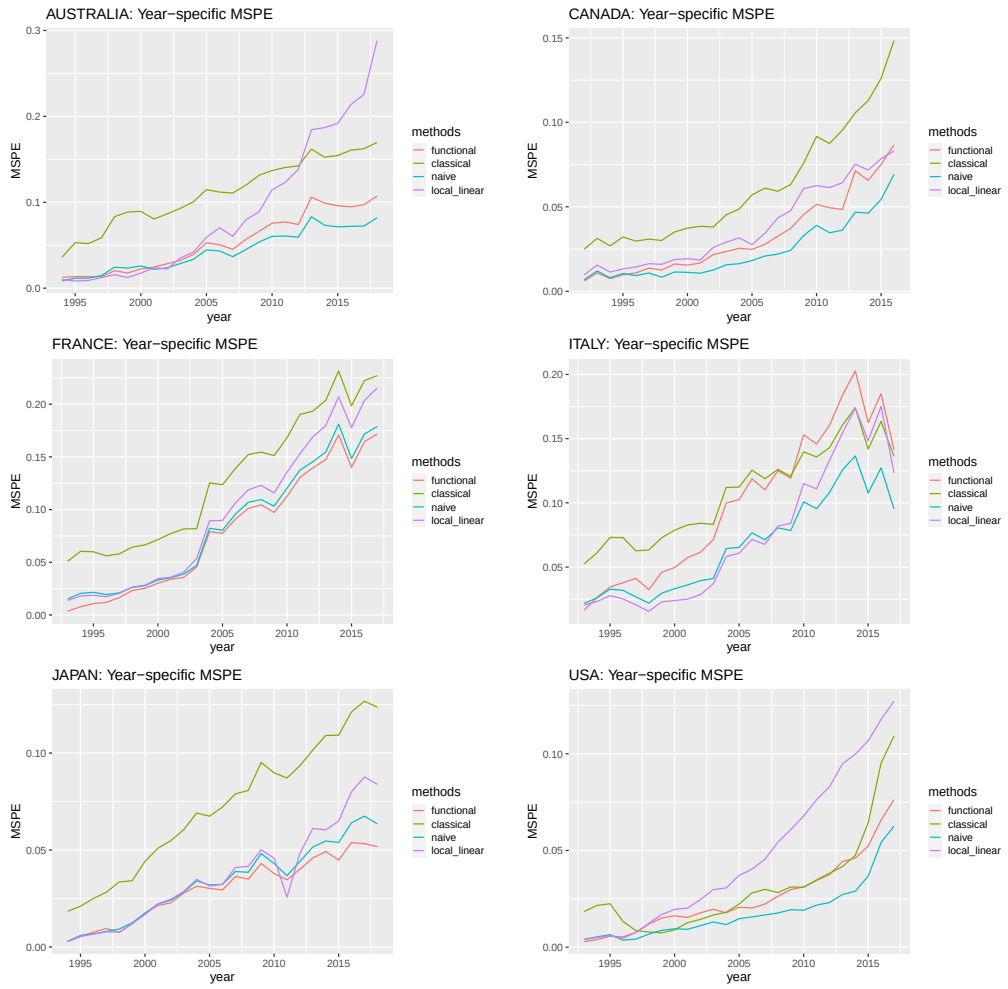


Figure 3.10: Year-specific MSPE by country and method (**functional**: functional data model, **classical**: classical factor model, **local regression**: time-varying factor model based on local linear regression, **naive**: time-varying factor model based on naive method); length of forecast horizon is 25 years

Australia and the US, it has worse forecasting performances than the classical factor model in the long term.

3.5.4 Estimate the optimal 'boundary'

From the empirical results above, we see that under the framework of time-varying model, the local regression method (by assuming factor loading will change in the future) is better at short-term forecasting while the naive method

(by assuming factor loading is in-variant in the future) is better at long-term forecasting. Therefore, we are interested in the boundary between short-term and long-term forecasting that divides the forecast horizon according to the predictive power of local regression method and the naive method.

By applying the estimation method introduced in Section 3.3, we estimate the optimal 'boundary' between the short-term and long-term forecasting, which is favored by the local regression method (time-varying forecast of the factor loading) and the naive method (time-invariant forecast of the factor loading) respectively. Recall that the optimal 'boundary' can be regarded as the optimal value of k defined in the hybrid forecasting method in Section 3.3. Applying the same datasets introduced in Section 3.4, we use the last 25 years of the historical data as the validation set, and the remaining data as the training set. In addition, to check the sensitivity of the least squares estimator to the division of validation set and testing set, we consider the last p years ($p = 15, 20, 25, 30$, respectively) of data as the validation set and the remaining data as the training set. Please refer to Appendix B.2 for more details

As defined in Section 3.3, for each set of data we compute the least squares estimator for the optimal 'boundary' in the forecasting horizon as

$$\widehat{k} = \underset{0 \le k \le T - T_0}{argmin}\ SSR(k),$$

where $k = 0, 1, 2, \ldots, T - T_0$. Here k represents the length of the short-term forecasting horizon, or the 'boundary' between short-term (based on the local regression method) and long-term (based on the naive method) forecasting. The local linear regression is used to make forecasts from $T_0 + 1$ to $T_0 + k$; while the naive method (i.e. keep the factor loading the same as that in time $T_0 + k$) is used to make forecasts from $T_0 + k + 1$ to T.

We plot the total SSR versus k in Figure 3.11. As shown in the plots, for each country, there is a minimum point of k corresponding to the smallest values

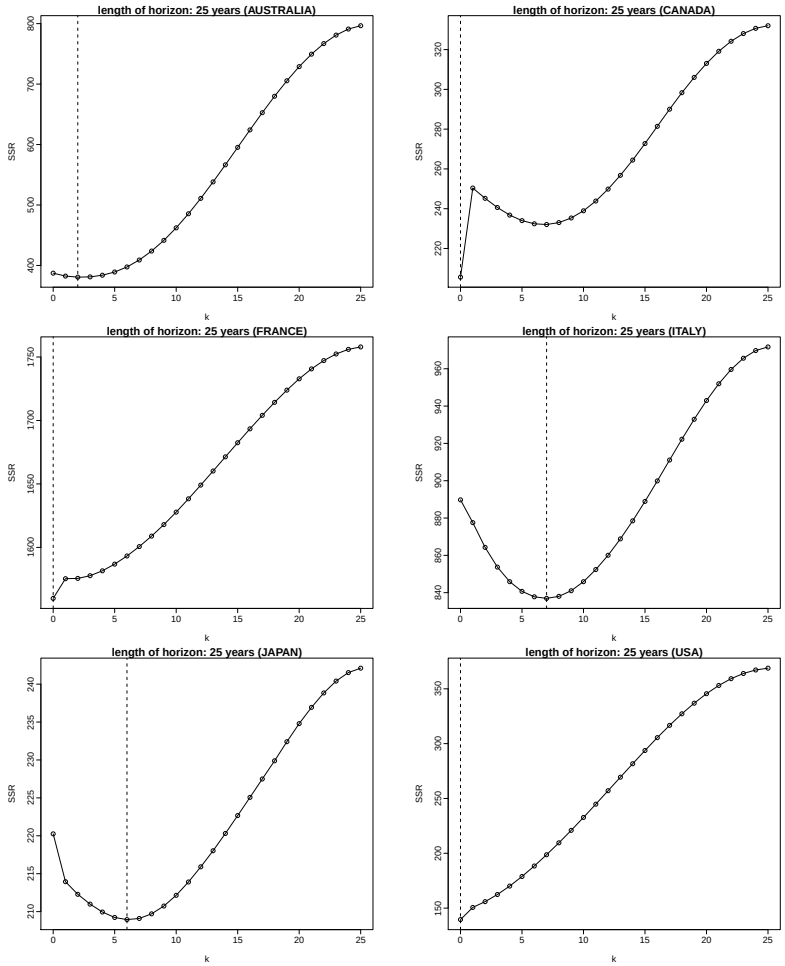


Figure 3.11: Plots of the total sum of squared residuals (SSR) versus the length (k) of the short-term forecast horizon (based on the hybrid forecasting method of time-varying factor model); length of forecast horizon: 25

of SSR, which indicates the optimal length of the short-term forecasting horizon for the time-varying factor model. For example, the plot of Italy shows that, when the value of k equals 7, SSR achieves the smallest value. Thus, based on the historical Italy mortality data, we suggest that it is better to use the local regression method for the short-term forecasting (less than 7 leads), as it puts more weights on the recent observations. As for forecasting more than 7 years ahead, the naive method generates more accurate predictions. However, as for US data, the optimal length of the short-term forecasting horizon is 0, which means there is no need to extrapolate factor loadings using local linear regression

and we should use the naive method alone. In Figure 3.11, we can observe jumps at $k = 1$ for some countries (such as Canada and France). When $k = 0$, the hybrid method is just the naive method and the extrapolation of factor loading is based on the historical estimation; while when $k > 1$, the hybrid method is based on the local regression method before the time $T_0 + k$. Therefore, a sudden change of SSR when k increases from 0 to 1 indicates that the pattern of factor loadings has a relatively large change after the time T_0.

Now we have observed the existence of positive optimal 'boundary' for some countries (such as Australia, Italy and Japan). Then based on the estimation of optimal 'boundary' k, we can compare the out-of-sample forecasting performances of the hybrid method with other methods. Here, we consider the Australian mortality data as an example since the estimation of the optimal boundary for Australian data is relatively stable (see details in Remark 3.4 and Appendix B.2).

Similar to previous analysis, we choose the last 25 years of the historical data as the testing set. And the remaining data is the training set. To estimate k, we use the last 25 years of the training data as the validation set to estimate the optimal 'boundary'. We then apply the estimation procedure in Section 3.3 to the training data and the forecasting method in the testing data using the estimated optimal 'boundary' to get the out-of-sample performance. The empirical results are shown in Figure 3.12 and Table 3.3.

In Table 3.3, we compute the overall MSPE over 1994 to 2018 for each forecasting method. We find that the time-varying model based on the hybrid forecasting method has the best out-of-sample performance among all four methods. And the naive method is a little bit worse than the hybrid method. Additionally, in Figure 3.12, we plot the year-specific MSPE for all different methods. For short-term forecasting, the hybrid method shows similar performance with the local regression method; while for the long-term forecasting, the hybrid method shows similar performance with the naive method. Therefore, the hybrid method is superior for both short-term and long-term forecasting, as it benefits from the

advantages of both methods. In practice, both naive method and hybrid method are recommended. The hybrid method could produce more accurate predictions while the naive method is much easier to implement.

Table 3.3: Australia: Overall MSPE over 1994 to 2018 (**Classical**: classical factor model, **TV-Local Regression**: time-varying factor model based on local regression method), **TV-Naive**: time-varying factor model based on naive method), **TV-Hybrid**: time-varying factor model based on hybrid method)

Country	Classical	TV-Local Regression	TV-Naive	TV-Hybrid
Australia	0.11162	0.08932	0.04342	**0.04288**

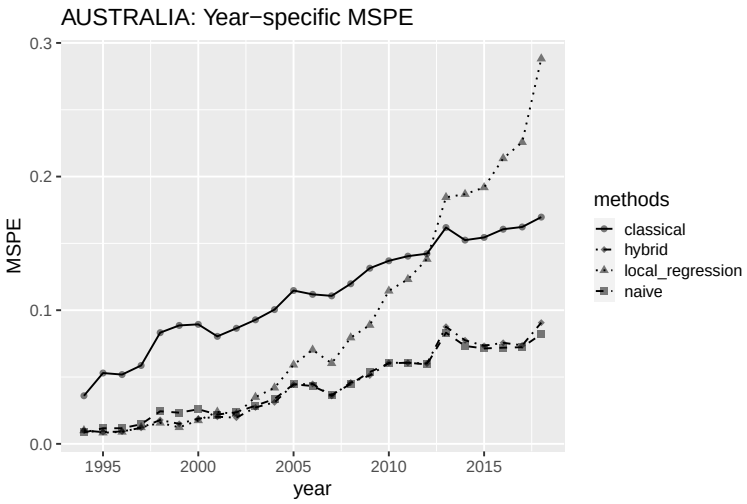


Figure 3.12: Australia: Year-specific MSPE over 1994 to 2018 (**classical**: classical factor model, **local regression**: time-varying factor model based on local regression method), **naive**: time-varying factor model based on naive method), **hybrid**: time-varying factor model based on hybrid method)

Remark 3.4. *In Appendix B.2, we further investigate the sensitivity of the optimal 'boundary' estimation by showing plots of the total sum of squared residuals (SSR) versus k using different lengths of training and testing datasets. For some countries, like Japan and USA, the least estimators are relatively stable. However, for countries like Canada and Italy, estimation of optimal 'boundary' is*

somewhat sensitive to the length of the dataset. This is reasonable as the time-varying model highly depends on the data and its intrinsic patterns. Therefore, when using different training data and forecasting horizons to estimate optimal 'boundary', we may obtain different values of k.

3.6 Monte carlo simulations

In this section, we further investigate the prediction performance of the time-varying factor model and the classical factor model through Monte Carlo simulations. We use examples with different structures of the factor loadings to illustrate that the time-varying factor model can improve the forecasting accuracy when the 'true' factor loadings change over time. In addition, we explain under which conditions the naive method performs better than the local regression method even in the short-term.

Similar to the previous empirical analysis, we denote the classical factor model as 'Classical' and the two forecasting methods based on the time-varying factor model as 'TV-Local Regression' method and 'TV-Naive' method, respectively. We show that when the 'true' factor loadings change over time, both forecasting methods based on the time-varying factor model outperform the 'Classical' method in forecasting. Moreover, the 'TV-Naive' method performs similarly to the 'TV-Local Regression' method for short-term forecasting; while it performs better than the 'TV-Local Regression' for long-term forecasting.

3.6.1 Data generating processes (DGP's)

We generate the centered data $x_{i,t}$ with one common factor:

$$x_{i,t} = b_{i,t} \cdot k_t + \epsilon_{i,t}, \quad i = 1, 2, \ldots, N,$$

where $k_t = k_{t-1} + w_t$. w_t follows independent identically distributed normal distributions, $N(0, 0.8^2)$. Thus the common factor k_t follows a random walk. We consider the following settings for different factor loadings $b_{i,t}$ and error terms $\epsilon_{i,t}$. In each setting below, we apply the normalization condition mentioned in Section 3.2, so that $b_{i,t}$ is normalized to sum to unity for each t.

- **DGP 1** (time-invariant factor loading):

$$b_{i,t} = b_i \sim i.i.d\ uniform(0,1),\ \epsilon_{i,t} \sim i.i.d\ N(0, 0.1^2).$$

- **DGP 2** (single-point structural change):
  For $i = 1, 2, \cdots, N/2$,

$$b_{i,t} = \begin{cases} b_i & for\ t = 1, 2, ..., T/2 \\ b_i + 1 & for\ t = T/2 + 1, ..., T \end{cases};$$

  and for $i = N/2 + 1, \cdots, N$,

$$b_{i,t} = \begin{cases} b_i & for\ t = 1, 2, ..., T/2 \\ b_i - 1 & for\ t = T/2 + 1, ..., T \end{cases};$$

$$b_i \sim i.i.d\ uniform(1.1, 1.9),\ \epsilon_{i,t} \sim i.i.d\ N(0, 0.03^2).$$

- **DGP 3** (continuous structural change):

$$b_{i,t} = \frac{1}{1 + e^{(\frac{6i}{N} + 2 - \frac{12t}{T})}},\ \epsilon_{i,t} \sim i.i.d\ N(0, 0.1^2).$$

DGP 1 follows the classical factor model with time-invariant factor loadings, and **DGP 2** and **DGP 3** exhibit different structures of the time-varying factor loadings. **DGP 2** describes a single-point structural change in the factor loadings; while **DGP 3** considers a continuous structural change in the factor loadings.

For each i, the factor loadings generated in **DGP 3** are monotonic functions and would converge to some constant as time goes by. The factor loadings in **DGP 3** may go up and down and would never diverge to extreme values, which is similar to the estimated time-varying factor loadings in Figure 3.3 using the US mortality data.

3.6.2 Comparison of the forecasting performance

To compare the forecasting performance of the time-varying factor model and the classical time-invariant factor model, we use the out-of-sample testing approach in the following analysis. For each DGP, we simulate 100 data sets with the dimension and sample sizes $N = T = 100$. For each data set, we consider the first k years of the data as the training set, and the remaining $T - k$ years of the data as the testing set ($k = 70, 75, 80, 85, 90, 95$, respectively). The model is firstly fitted using the training set, and then forecasted in the testing set. We employ the mean squared prediction error (MSPE) as the measure to evaluate performance of different models.

Table 3.4 reports the comparison results based on various lengths of the training and testing sets. An example of the estimated and forecasted factor loadings is showen in Figure 3.13 to better explain the results. As shown in Table 3.4, the two time-varying methods perform worse than the classical factor model for **DGP 1**, which assumes the time-invariant factor loadings. The less accurate prediction results can be attributed to the inaccurate estimation from the time-varying model, which is supported by the left plot of Figure 3.13. When the 'true' factor loadings are time-invariant, the estimation from the time-varying model goes up and down randomly due to the over-fitting problem of the non-parametric estimating method. Therefore, the forecasting based on these estimation is not satisfied. However, the classical method provides a close estimation and better forecasting in this case.

Table 3.4: Comparison of forecasting performance of the time-varying factor model and the classical factor model (based on the different lengths of training sets)

DGPs	methods	70	75	80	85	90	95
DGP 1	TV-Local Regression	0.2300	0.1873	0.1249	0.0852	0.0590	0.0344
	TV-Naive	0.2239	0.1849	0.1228	0.0837	0.0582	0.0342
	Classical	**0.2209**	**0.1839**	**0.1222**	**0.0829**	**0.0575**	**0.0336**
DGP 2	TV-Local Regression	0.2597	0.2046	0.1230	0.0818	0.0528	0.0265
	TV-Naive	**0.2291**	**0.1874**	**0.1227**	**0.0817**	**0.0527**	**0.0265**
	Classical	0.2542	0.2121	0.1482	0.0946	0.0643	0.0372
DGP 3	TV-Local Regression	0.1757	0.1384	0.0987	0.0737	0.0506	0.0365
	TV-Naive	**0.1704**	**0.1319**	**0.0940**	**0.0724**	**0.0499**	**0.0362**
	Classical	0.2078	0.1759	0.1456	0.1162	0.0917	0.0748

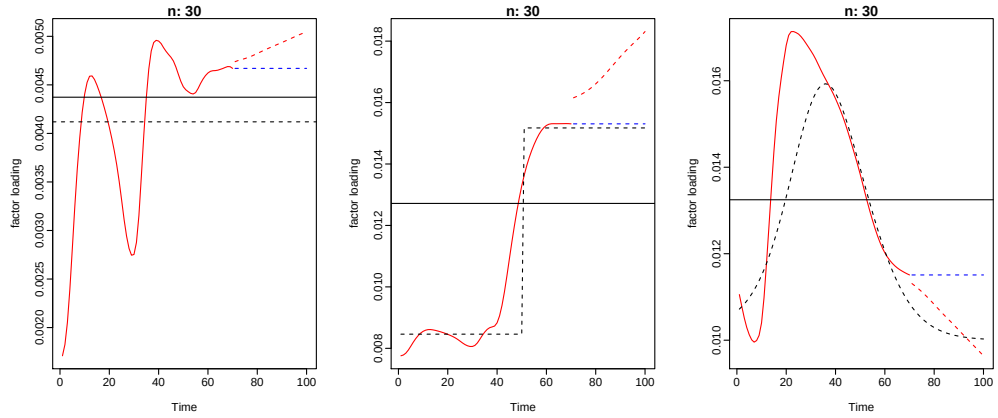


Figure 3.13: Comparison of the factor loadings: estimation and forecast. From left to right: **DGP 1**, **DGP 2**, **DGP 3**. $k = 70$. Black dashed line: true factor loadings. Black solid line: estimation from the classical factor model ('Classical'). Red solid line: estimation from the time-varying factor model. Red dashed line: 'TV-Local Regression'. Blue dashed line: 'TV-Naive'.

DGP 2 and **DGP 3** follow the structures of time-varying factor models with abrupt and continuous changes in the factor loadings, respectively. From Table 3.4, we see that both the two time-varying methods perform better than the classical method when the 'true' factor loadings are time-varying. In particular, the naive method performs the best in these two cases, especially for long-term forecasting. The superior forecasts of the two time-varying methods result from the more accurate estimation of the time-varying factor loadings, which can be seen

from the middle and right plots in Figure 3.13. We find that the classical model cannot capture the changed factor loadings as it assumes the factor loadings are time-invariant, while the time-varying model can estimate those changing factor loadings accurately and provides a solid foundation for the forecasting step.

Further analysis of Table 3.4 suggests that the local regression method and the naive method preform similarly for the relatively short-term forecasting ($k = 85, 90, 95$), while for long-term forecasting ($k = 30, 25$), the naive method performs better. This phenomenon can be explained by the plot of **DGP 3** in Figure 3.13. From the plot, we see that the forecast of the local regression method follows the trend of the estimated factor loadings. Therefore, when the trend remains the same in a short period of time, the forecast of the local regression method is satisfying. However, as the trend of the 'true' factor loadings changes, the forecast of the local regression method diverges away from the true values in the long term. This is the drawback of the non-parametric forecasting method. On the other hand, the constant factor loadings used in the forecasting procedure of the naive method guarantee that the factor loadings will not diverge dramatically and result in a better performance for long-term forecasting. In addition, although the 'true' factor loadings in the training set changes, if it is time-invariant in the forecasting horizon, the naive method is also better than the local regression method even in the short term (which is supported by the plot of **DGP 2** in Figure 3.13). The aforementioned analysis could be used to explain why the estimated optimal 'boundary' of the US data is 0 in Section 3.5.4. From Figure 3.3 we see that in the first years of the forecasting horizon, the trends of the factor loadings are either different from that in the training set or remain flat. However, the local regression method cannot capture the unknown changing trends, so the naive method can outperform it even in the short term. In this case, the naive method not only captures the time-varying factor loadings in the estimation step, but also uses the constant factor loadings in the forecasting step, so it is overall preferable when the factor loadings are changing over time.

3.7 Conclusion

There is a vast literature using factor models for mortality modelling and forecasting, such as the Lee-Carter model and its variants. Factor loadings, which capture the relationship between different variables and the latent common factors, are usually assumed to be time-invariant over time, which is too restrictive in reality. In this chapter, we develop a time-varying factor model for mortality modelling and two corresponding forecasting methods, which can improve the forecasting performance compared with using the classical factor model. To understand the optimal forecasting horizon of the two forecasting methods based on the time-varying factor model, we propose a method to estimate the optimal 'boundary' between short-term and long-term forecasting, which is favoured by the local regression method (time-varying forecast of the factor loading) and the naive method (time-invariant forecast of the factor loading), respectively.

In addition, we introduce the estimation and forecasting methods of the time-varying factor model. To make out-of-sample forecasting, we consider modelling and extrapolating the common factors and factor loadings separately. The common factor is modelled and forecasted using the ARIMA model; while the factor loadings are estimated and extrapolated using the local linear regression method or the naive method. By estimating the optimal boundary between short-term and long-term forecasting, we propose the hybrid forecasting method. Based on these methods, we can forecast mortality rates into the future with the time-varying factor model. Multiple countries' mortality data are used for empirical analysis. We find that the time-varying factor model provide superior out-of-sample forecasting performance. Using simulation studies, we show the performance of the time-varying factor model and classical factor model under different scenarios and explain why the naive method performs better than the local regression method.

Chapter 4

Robust Principal Component Analysis for High Dimensional Data Based on Characteristic Transformation

4.1 Introduction

High-dimensional data are ubiquitously encountered with the fast development of modern technologies (Donoho [2000], Johnstone and Titterington [2009], Lee et al. [2014], Morales-Jimenez et al. [2018], etc.). Examples include genomic data, financial data, and medical image data. The data often have millions of features with comparable or relatively small sample sizes. With the explosion of the dimension, the heterogeneity, which is defined as the diversity of statistical properties of the data, becomes more and more common. Not only the quantity of the heterogeneity is larger, but also the styles of it are more various. For example, features with heavy-tailed distribution are more likely to present along with the normal distributed features in high dimensional data. Other types of heterogeneity includes heteroscedastic noise, unknown nonlinearity, and outliers. As traditional statistical assumptions are always violated due to these heterogeneities, it is of great urgency to develop new approaches and theories for the

high-dimensional regime.

For the high-dimensional data, the principal component analysis (PCA) is a widely used technique for data exploration and dimension reduction (Anderson [2003], Jolliffe [2002]). The idea is to search for low-dimensional projections that can represent the high-dimensional data. Intuitively PCA intends to pursue a small number of common features possessed by all variables under study. Mathematically the classical PCA is based on the covariance matrix, and the leading eigenvectors of the sample covariance matrix serve as the directions of the projections. Unfortunately, the sample covariance matrix is very sensitive to the heterogeneities in the data (Li and Chen [1985]), so the standard PCA is not robust, especially under the current high-dimensional regime. The non-robustness issue of PCA has been studied in robust statistical analysis. A natural and simple idea is to replace the sample covariance matrix with a robust estimator. Croux and Haesbroeck [2000] studied the influence functions and efficiencies of some robust covariance matrix estimators. Another approach, using a projection-pursuit index instead of the variance to measure the dispersion of the projections, is proposed by Li and Chen [1985]. More recently, studying PCA in the view of a low-rank matrix approximation problem and minimizing the robust loss function has attracted attention in computer science (Candès et al. [2011]). See, for example, Vidal et al. [2016], Cui et al. [2003] and She et al. [2016] for more reviews.

As mentioned before, the styles of heterogeneity are diverse under the high dimensional regime. Hence, we are not only interested in dealing with the samples contaminated by typical outliers but also the data drawing from heavy-tailed distributions. Imagine that the population distributions of the data have infinite second moments or even infinite first moments, then any method depends on those moments, such as the standard PCA and the robust-covariance-based PCA, are invalid. Motivated by this difficulty, we propose a novel robust PCA, in which the pivotal step is transforming the original data based on

the form of the characteristic function. Our proposed method is robust to different styles of heterogeneity, even to data with infinite population moments. The robustness mainly comes from the appealing properties of the transformation. Recall that for a real-valued random variable y, its characteristic function is $\phi(t) = \boldsymbol{E}(exp\{\mathrm{i}ty\})$ $(t \in \mathbb{R}, \mathrm{i}^2 = -1)$, which completely defines the probability distribution of y and $|exp\{\mathrm{i}ty\}| = 1$ for any t. Hence the transformation $z_i = exp\{\mathrm{i}y_i\}$ $(i = 1, \ldots, p)$ retains the distribution information of y_i $(i = 1, \ldots, p)$, and more noteworthily is a bounded random variable no matter y_i $(i = 1, \ldots, p)$ is bounded or not. As a result, the standard PCA is valid on the transformed variable z_i $(i = 1, \ldots, p)$ as the transformation shrinks the effect of extreme outliers and also guarantees a finite second moment. Moreover, due to the nonlinear nature of the exponential function, the transformation helps explore the nonlinear relationship in $y_i (i = 1, \ldots, p)$ and it can be regarded as a special case of Kernel PCA (Chapter 4.1 in Vidal et al. [2016]), the algorithm of which can benefit the computation when the dimension is extremely large.

Apart from the heterogeneity, the high dimensionality itself is a crucial problem in classical statistics. Donoho [2000] discussed the curse and blessing of the high dimensionality in a wide range of statistical problems. Johnstone [2001]; Lam et al. [2011]; Lee et al. [2014]; Wang and Fan [2017]; Cai et al. [2017] and others have made effort to understand the behavior of the empirical eigenvalues under the high dimensional setting when the sample size n and the dimension p both go to infinity. It is well known from these literature that, the stronger spikeness of the population and the larger sample size allow larger dimension in consistently recovering the population eigenvalues from the empirical eigenvalues. Fortunately, as our proposed method provides bounded variables after the transformation, many of the theorems from those literature can be applied to our proposed method to generate interesting results. Specifically, we assume a spiked covariance model, according to Wang and Fan [2017] and Cai et al. [2017], on the transformed data, to study the properties of our proposed method under the high

dimensional setting. Two aspects of statistical properties are studied in Section 4.3. Firstly, a general upper bound for the excess error (the difference between the optimal (population) reconstruction error and the empirical reconstruction error) of the PCA methods is given in Section 4.3.1, which shows the robust PCA can always achieve small excess errors while the standard PCA may not. Secondly, the behavior of the largest k eigenvalues is investigated. Specifically, we analyse the spiked structure for both of original data and transformed data, as well as the estimation biases of their empirical eigenvalues due to curse of dimensionality. Through the analysis, we find that the transformation retains the spiked structure but shrinks the leading eigenvalues, which makes the eigenvalues that are mixed with non-spiked ones more biased. However, for the heavy-tail-distributed data, the empirical spiked eigenvalues of the transformed data are generally closer to the population ones than those of the original data.

In addition, our proposed method can be used to reconstruct the original data. We illustrate the advantage of our proposed method against the classic PCA in the sense of mean squared reconstruction error (MSE) with several examples. Those examples include data with heterogeneity in variances, data with outliers, and data from three different heave-tailed distributions. In total, we find that our proposed method can recover those data more accurately than the classic PCA.

At last, we demonstrate an example of applying the method in real data analysis by analyzing the protein expression measurements of mice from Higuera et al. [2015]. Most of the proteins have heavy tails or extreme outliers in their expression levels, so it is essential to use robust methods on the data. The proposed method is used to classify mice with different genotypes based on their protein expression data. Comparing to the classical PCA, our proposed method can identify the mice with abnormal genotype more accurately.

The rest of this chapter is organized as follows. Section 4.2 describes our proposed method in details. Section 4.3 studies the statistical properties. Sim-

ulations to illustrate the reconstruction performance under different cases are presented in Section 4.4. Section 4.5 gives an example of a real data application.

4.2 Methodology

Let us recall the settings of the classical Principal Component Analysis (PCA). Suppose we have n data points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, generated by a random vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_p)^\top \in \mathbb{R}^p$. The classical PCA aims to find a subspace $S \subset \mathbb{R}^p$ of dimension k $(k < p)$ that best fits those data points. Mathematically, the problem can be written as the following optimization problem:

$$\min_{\boldsymbol{u}, \boldsymbol{U}, \{\boldsymbol{x}_i\}} \sum_{i=1}^{n} \|\boldsymbol{y}_i - \boldsymbol{u} - \boldsymbol{U}\boldsymbol{x}_i\|^2 \quad \text{s.t.} \quad \boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_k \quad \text{and} \quad \sum_{i=1}^{n} \boldsymbol{x}_i = \boldsymbol{0},$$

where $\boldsymbol{u}$ is a point which represents the central of the subspace, $\boldsymbol{U}$ is a $p \times k$ matrix whose columns are the basis of the subspace and $\boldsymbol{x}_i \in \mathbb{R}^k$ is the vector of the new coordinates of $\boldsymbol{y}_i$ in the subspace. The optimal solution to the standard PCA (Chapter 2.1.2 in Vidal et al. [2016]) can be obtained as

$$\widehat{\boldsymbol{u}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_i \quad \text{and} \quad \widehat{\boldsymbol{x}}_i = \widehat{\boldsymbol{U}}^\top \left(\boldsymbol{y}_i - \widehat{\boldsymbol{u}}\right),$$

where $\widehat{\boldsymbol{U}}$ is a $p \times k$ matrix whose columns are the eigenvectors corresponding to the the largest k eigenvalues of the sample covariance matrix

$$\widehat{\boldsymbol{\Sigma}}_n = \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{y}_i - \widehat{\boldsymbol{u}}\right) \left(\boldsymbol{y}_i - \widehat{\boldsymbol{u}}\right)^\top.$$

Then $\widehat{\boldsymbol{U}}\widehat{\boldsymbol{x}}_i$ is the low-rank approximation of $\boldsymbol{y}_i$ if we assume $\mathbb{E}(\boldsymbol{y}) = \boldsymbol{0}$ and $\widehat{\boldsymbol{u}} = \boldsymbol{0}$ without lose of generality. However, it is well know that if the data contains extreme values or have a heavy-tailed distribution, the above optimization is not reliable and the solution $\widehat{\boldsymbol{U}}\widehat{\boldsymbol{x}}_i$ is not a good low-rank approximation. For example,

consider the data points coming from a heavy-tailed distribution without a finite second moment, then the covariance matrix $\widehat{\boldsymbol{\Sigma}}_n$ will be extremely unreliable and invalid to make inferences on the population covariance matrix.

To address this issue, we propose a new PCA method to obtain a good approximation of $\boldsymbol{y}$, which is robust to outliers and heavy-tailed distributions in this chapter. The idea is to find a transformation which is robust to the heavy-tailed distribution or extreme values of the original data and then conduct the classical PCA on the transformed data instead. The details of the method is described as follows.

Let $\boldsymbol{z} = (z_1, z_2, \ldots, z_p)^\top$ be the transformed data of $\boldsymbol{y}$, where the transformation is $z_j = e^{\mathrm{i}y_j}$ $(j = 1, 2, \ldots, p)$ and i is the imaginary unit. The reasons to make this transformation come from the special properties of $\boldsymbol{z}$. Firstly, $\boldsymbol{z}$ has finite second moments and contains most of the information in $\boldsymbol{y}$ as it has the form of the characteristic function of $\boldsymbol{y}$, which solves the problem that $\boldsymbol{y}$ comes from heavy-tailed distributions, especially for those without the second moments. Secondly, the mode of z_j equals 1 for any $j = 1, \ldots, p$, which means the variance of it is bounded. This property shrinks the effect of the possible outliers or extremely various variances on the result of the dimension reduction. Thirdly, due to the non-linear property of the transformation, it is capable of revealing the non-linear relationship between components in $\boldsymbol{y}$ unlike the classical PCA, which can only detect the linear relationships.

While there are desired properties with $\boldsymbol{z}$, it contains complex elements which make the situation complicated. On the other hand, according to Euler's formula, z_j can be written as:

$$z_j = e^{\mathrm{i}y_j} = \cos y_j + \mathrm{i} \sin y_j \quad (j = 1, \ldots, p).$$

Then if we define $\boldsymbol{r} = \Big(\cos y_1, \ldots, \cos y_p, \sin y_1, \ldots, \sin y_p\Big)^\top$, we have $\boldsymbol{z}$ as a

linear transform of $\boldsymbol{r}$:

$$\boldsymbol{z} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \mathrm{i} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \mathrm{i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & \mathrm{i} \end{pmatrix} \begin{pmatrix} \cos y_1 \\ \vdots \\ \cos y_p \\ \sin y_1 \\ \vdots \\ \sin y_p \end{pmatrix} := \begin{pmatrix} \boldsymbol{I}_p & \mathrm{i}\boldsymbol{I}_p \end{pmatrix} \boldsymbol{r}.$$

Assume there also exists a low rank subspace which best fits data points $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ generated from $\boldsymbol{z}$. Then, to find a good low-rank approximation of $\boldsymbol{z}_i (i = 1, \ldots, n)$, we only need conduct the classical PCA on $\boldsymbol{r}_i (i = 1, \ldots, n)$, which is real valued.

Suppose $\boldsymbol{\Sigma}_{\boldsymbol{r}}$ is the covariance matrix of $\boldsymbol{r}$, and $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_k$ are the orthonormal eigenvectors corresponding to the k largest eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_k$ of $\boldsymbol{\Sigma}_{\boldsymbol{r}}$. By the classical PCA method, $\boldsymbol{r}$ is approximated by

$$\begin{aligned} \widetilde{\boldsymbol{r}} &= \mathbb{E}(\boldsymbol{r}) + \sum_{j=1}^{k} \boldsymbol{\beta}_j \boldsymbol{\beta}_j^{\top} \left(\boldsymbol{r} - \mathbb{E}(\boldsymbol{r})\right) \\ &= \mathbb{E}(\boldsymbol{r}) + \left(\sum_{j=1}^{k} \left(\boldsymbol{r} - \mathbb{E}(\boldsymbol{r})\right)^{\top} \boldsymbol{\beta}_j \boldsymbol{\beta}_j^{(\cos)\top}, \sum_{j=1}^{k} \left(\boldsymbol{r} - \mathbb{E}(\boldsymbol{r})\right)^{\top} \boldsymbol{\beta}_j \boldsymbol{\beta}_j^{(\sin)\top} \right)^{\top}, \end{aligned}$$

where $\boldsymbol{\beta}_j^{(\cos)} = (\beta_{1,j}, \beta_{2,j}, \ldots, \beta_{p,j})^{\top}$ and $\boldsymbol{\beta}_j^{(\sin)} = (\beta_{p+1,j}, \beta_{p+2,j}, \ldots, \beta_{2p,j})^{\top}$, which are the first half and the second half of $\boldsymbol{\beta}_j$, respectively.

Therefore, the low-dimensional approximation of $\boldsymbol{z}$ is

$$\begin{aligned} \widetilde{\boldsymbol{z}} &= \begin{pmatrix} \boldsymbol{I}_p & \mathrm{i}\boldsymbol{I}_p \end{pmatrix} \widetilde{\boldsymbol{r}} \\ &= \begin{pmatrix} \boldsymbol{I}_p & \mathrm{i}\boldsymbol{I}_p \end{pmatrix} \mathbb{E}(\boldsymbol{r}) + \\ &\begin{pmatrix} \boldsymbol{I}_p & \mathrm{i}\boldsymbol{I}_p \end{pmatrix} \left(\sum_{j=1}^{k} \left(\boldsymbol{r} - \mathbb{E}(\boldsymbol{r})\right)^{\top} \boldsymbol{\beta}_j \boldsymbol{\beta}_j^{(\cos)\top}, \sum_{j=1}^{k} \left(\boldsymbol{r} - \mathbb{E}(\boldsymbol{r})\right)^{\top} \boldsymbol{\beta}_j \boldsymbol{\beta}_j^{(\sin)\top} \right)^{\top} \end{aligned}$$

$$= \mathbb{E}(\boldsymbol{z}) + \sum_{j=1}^{k} \boldsymbol{\beta}_j^{(\cos)} \boldsymbol{\beta}_j^\top (\boldsymbol{r} - \mathbb{E}(\boldsymbol{r})) + \mathrm{i} \sum_{j=1}^{k} \boldsymbol{\beta}_j^{(\sin)} \boldsymbol{\beta}_j^\top (\boldsymbol{r} - \mathbb{E}(\boldsymbol{r})) .$$

With data points $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n$, it is straightforward to estimate $\mathbb{E}(\boldsymbol{r})$, $\mathbb{E}(\boldsymbol{z})$ and $\boldsymbol{\Sigma}_{\boldsymbol{r}}$ by

$$\overline{\boldsymbol{r}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{r}_i, \quad \overline{\boldsymbol{z}} = \begin{pmatrix} \boldsymbol{I}_p & \mathrm{i}\boldsymbol{I}_p \end{pmatrix} \overline{\boldsymbol{r}}, \quad \text{and} \quad \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{r},n} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{r}_i - \overline{\boldsymbol{r}}) (\boldsymbol{r}_i - \overline{\boldsymbol{r}})^\top ,$$

respectively. In addition, estimate $\boldsymbol{\beta}_j$ $(j = 1, \ldots k)$ by the eigenvectors $\widehat{\boldsymbol{\beta}}_j$ $(j = 1, \ldots k)$ of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{r},n}$. The method to estimate k can be various and we use the accumulative variance as the criterion in the empirical analysis for simplicity. Other reasonable criterion can be applied under different purposes.

Finally, to recover the original data, we only need to transform back from $\widetilde{\boldsymbol{z}}_i (i = 1, \ldots, n)$. The approximation of $\boldsymbol{y}_i (i = 1, \ldots, n)$ is:

$$\begin{aligned} \widetilde{\boldsymbol{y}}_i &= \frac{1}{\mathrm{i}} \log(\widetilde{\boldsymbol{z}}_i) + 2h_i\pi\mathbf{1} \\ &= \frac{1}{\mathrm{i}} \log \left(\overline{\boldsymbol{z}} + \sum_{j=1}^{k} \widehat{\boldsymbol{\beta}}_j^{(\cos)} \widehat{\boldsymbol{\beta}}_j^\top (\boldsymbol{r}_i - \overline{\boldsymbol{r}}) + \mathrm{i} \sum_{j=1}^{k} \widehat{\boldsymbol{\beta}}_j^{(\sin)} \widehat{\boldsymbol{\beta}}_j^\top (\boldsymbol{r}_i - \overline{\boldsymbol{r}}) \right) + 2h_i\pi\mathbf{1}, \end{aligned} \tag{4.2.1}$$

where $\log(\boldsymbol{a}) = (\log(a_1), \log(a_2), \ldots, \log(a_n))^\top$ for any n-dimensional vector $\boldsymbol{a}$, and $h_i (i = 1, \ldots, n)$ is an integer which needs to be estimated in practice.

Remark 4.1. *The computational algorithm is summarized in Algorithm 3. Note that $\widetilde{\boldsymbol{z}}_i (i = 1, \ldots, n)$ consists of complex numbers. The complex logarithm can have infinite many values, due to the periodicity of the complex exponential function. According to Euler's formula, those values are different by multiples of $2h\pi$. Therefore, in equation (4.2.1), we need to find the h_i to ensure that $\widetilde{\boldsymbol{y}}_i$ is a good approximation to $\boldsymbol{y}_i$. Hence, in practice, we estimate h_i for each data point $\boldsymbol{y}_i$ by*

$$\arg\min_{h_i \in \mathbb{Z}} \left(\boldsymbol{y}_i - \left(\frac{1}{\mathrm{i}} \log(\widetilde{\boldsymbol{z}}_i) + 2h_i\pi\mathbf{1} \right) \right) \quad (i = 1, \ldots, n).$$

Algorithm 3: Robust PCA for High Dimensional Data

Input: Data $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n] \in \mathbb{R}^{p\times n}$; Desired rank $\leq p$.
Output: Low-dimensional representation of $\boldsymbol{Y}$.
Transformation Step:
Compute $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n] \in \mathbb{R}^{(2p)\times n}$, where $\boldsymbol{r}_i = (\cos y_{1,i}, \ldots, \cos y_{p,i}, \sin y_{1,i}, \ldots, \sin y_{p,i})^\top$;
PCA Step:
Compute the sample mean $\overline{\boldsymbol{r}} = n^{-1}\sum_{i=1}^n \boldsymbol{r}_i$;
Compute the sample variance-covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{r},n} = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{r}_i - \overline{\boldsymbol{r}})(\boldsymbol{r}_i - \overline{\boldsymbol{r}})^\top$;
Conduct eigendecomposition on $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{r},n}$ and get $\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_{\widehat{k}}$, the eigenvectors corresponding to the largest $\widehat{k}$ eigenvalues of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{r},n}$;
Inverse Transformation Step:
Compute $\widehat{\boldsymbol{\beta}}_j^{(\cos)} = (\beta_{1,j}, \ldots, \beta_{p,j})^\top$ and $\widehat{\boldsymbol{\beta}}_j^{(\sin)} = (\beta_{p+1,j}, \ldots, \beta_{2p,j})^\top$;
Compute $\overline{\boldsymbol{z}} = \begin{pmatrix}\boldsymbol{I}_p & \mathrm{i}\boldsymbol{I}_p\end{pmatrix}\overline{\boldsymbol{r}}$;
Compute $\widetilde{\boldsymbol{z}}_i = \overline{\boldsymbol{z}} + \sum_{j=1}^{\widehat{k}} \widehat{\boldsymbol{\beta}}_j^{(\cos)}\widehat{\boldsymbol{\beta}}_j^{(\cos)\top}(\boldsymbol{r}_i - \overline{\boldsymbol{r}}) + \mathrm{i}\sum_{j=1}^{\widehat{k}} \widehat{\boldsymbol{\beta}}_j^{(\sin)}\widehat{\boldsymbol{\beta}}_j^{(\sin)\top}(\boldsymbol{r}_i - \overline{\boldsymbol{r}})$;
Compute $\widehat{h}_i = \arg\min_{h_i\in\mathbb{Z}} \left(\boldsymbol{y}_i - \left(\frac{1}{\mathrm{i}}\log(\widetilde{\boldsymbol{z}}_i) + 2h_i\pi\mathbf{1}\right)\right) \quad (i = 1, \ldots, n)$;
Compute $\widetilde{\boldsymbol{y}}_i = \log(\widetilde{\boldsymbol{z}}_i)/\mathrm{i} + 2\widehat{h}_i\pi\mathbf{1}$, $i = 1, \ldots, n$.

Remark 4.2. *Our proposed method can be viewed as a special kind of nonlinear and Kernel PCA (Chapter 4.1 Vidal et al. [2016]). The nonlinear transformation is* $\phi(\cdot) : \mathbb{R}^p \to \mathbb{R}^{2p}$, *where* $\phi(\boldsymbol{y}) = (\cos y_1, \ldots, \cos y_p, \sin y_1, \ldots, \sin y_p)^\top$ *and the kernel function is*

$$\kappa(\boldsymbol{y}_i, \boldsymbol{y}_j) = \phi(\boldsymbol{y}_i)^\top\phi(\boldsymbol{y}_j) = \sum_{m=1}^p \cos y_{mi}\cos y_{mj} + \sum_{m=1}^p \sin y_{mi}\sin y_{mj}.$$

This shows that our method is capable to explore the non-liner relationship among the original data. In addition, we can compute the principal components with the kernel function described above according to the Kernel PCA algorithm (see Algorithm 4.1 in Vidal et al. [2016] for example), which is particularly useful when the dimension p is too large to compute the covariance matrix of the transformed data.

4.3 Statistical properties

We study the statistical properties of the robust PCA in this section. Firstly, we give a general upper bound for the excess errors (the difference between the optimal (population) reconstruction error and the empirical reconstruction error) of the PCA methods. It shows that due to the finite variance of the transformed data, the robust PCA can always achieve small excess errors. The standard PCA, however, will have extreme large excess error when the original data is heavy-tailed or contains outliers. We also discuss under which conditions the excess error is asymptotically zero. Secondly, the behavior of the largest k eigenvalues is studied based on a spiked covariance model. We find that the transformation in our proposed method shrinks the spiked structure while gives relatively better estimation of the spiked eigenvalues when the data is heavy-tailed (Table 4.1, 4.2, and 4.3).

4.3.1 The upper bound of the excess error

In this section, we give the upper bound for the aforementioned excess error in Theorem 4.1. The Theorem 4.1 is generally hold for both the standard PCA and the newly proposed robust PCA. Through this theorem we show that, under some conditions, the proposed method can achieve small excess error. The standard PCA, however, may fail to do that under the same conditions.

Let us first introduce some notations and definitions in order to illustrate the results. Suppose a random vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_p)^\top \in \mathbb{R}^p$ has mean $\mathbf{0}$ and variance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ are n independent samples of $\boldsymbol{y}$ and the corresponding sample covariance matrix is $\widehat{\boldsymbol{\Sigma}}$. Let $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p$ be the orthonormal eigenvectors corresponding to the eigenvalues of $\boldsymbol{\Sigma}$ in descending order, and $\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_p$ be those of $\widehat{\boldsymbol{\Sigma}}$. Denote $\boldsymbol{B}_k = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)$ and $\widehat{\boldsymbol{B}}_k = (\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_k)$ ($k < p$ is fixed). With $\boldsymbol{B}_k$, the basis of the optimal low-dimensional subspace,

we have

$$\boldsymbol{y} = \sum_{i=1}^{k} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y} + \sum_{i=k+1}^{p} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y} = \boldsymbol{B}_k \boldsymbol{B}_k^\top \boldsymbol{y} + \boldsymbol{u}(\boldsymbol{B}_k);$$
$$\boldsymbol{y}_j = \sum_{i=1}^{k} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y}_j + \sum_{i=k+1}^{p} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y}_j = \boldsymbol{B}_k \boldsymbol{B}_k^\top \boldsymbol{y}_j + \boldsymbol{u}_j(\boldsymbol{B}_k) \quad (j = 1, \ldots, n).$$

Similarly with $\widehat{\boldsymbol{B}}_k$, the empirical counterpart of $\boldsymbol{B}_k$, we have

$$\boldsymbol{y} = \sum_{i=1}^{k} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y} + \sum_{i=k+1}^{p} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y} = \widehat{\boldsymbol{B}}_k \widehat{\boldsymbol{B}}_k^\top \boldsymbol{y} + \boldsymbol{u}(\widehat{\boldsymbol{B}}_k);$$
$$\boldsymbol{y}_j = \sum_{i=1}^{k} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y}_j + \sum_{i=k+1}^{p} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y}_j = \widehat{\boldsymbol{B}}_k \widehat{\boldsymbol{B}}_k^\top \boldsymbol{y}_j + \boldsymbol{u}_j(\widehat{\boldsymbol{B}}_k) \quad (j = 1, \ldots, n).$$

Therefore, the (true) reconstruction error with $\boldsymbol{B}_k$ and $\widehat{\boldsymbol{B}}_k$ can be written as

$$R(\boldsymbol{B}_k) = \boldsymbol{E}\left((\boldsymbol{y} - \sum_{i=1}^{k} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y})^\top (\boldsymbol{y} - \sum_{i=1}^{k} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y}) \right) = \boldsymbol{E}\left(\boldsymbol{u}(\boldsymbol{B}_k)^\top \boldsymbol{u}(\boldsymbol{B}_k) \right);$$
$$R(\widehat{\boldsymbol{B}}_k) = \boldsymbol{E}\left((\boldsymbol{y} - \sum_{i=1}^{k} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y})^\top (\boldsymbol{y} - \sum_{i=1}^{k} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y}) \right) = \boldsymbol{E}\left(\boldsymbol{u}(\widehat{\boldsymbol{B}}_k)^\top \boldsymbol{u}(\widehat{\boldsymbol{B}}_k) \right).$$

The difference $R(\widehat{\boldsymbol{B}}_k) - R(\boldsymbol{B}_k)$ is the so-called (true) excess error. Furthermore, the corresponding empirical reconstruction errors are

$$\begin{aligned} R_n(\boldsymbol{B}_k) &= \frac{1}{n} \sum_{j=1}^{n} \left((\boldsymbol{y}_j - \sum_{i=1}^{k} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y}_j)^\top (\boldsymbol{y}_j - \sum_{i=1}^{k} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \boldsymbol{y}_j) \right) \\ &= \frac{1}{n} \sum_{j=1}^{n} \left(\boldsymbol{u}_j(\boldsymbol{B}_k)^\top \boldsymbol{u}_j(\boldsymbol{B}_k) \right); \\ R_n(\widehat{\boldsymbol{B}}_k) &= \frac{1}{n} \sum_{j=1}^{n} \left((\boldsymbol{y}_j - \sum_{i=1}^{k} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y}_j)^\top (\boldsymbol{y}_j - \sum_{i=1}^{k} \widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i^\top \boldsymbol{y}_j) \right) \\ &= \frac{1}{n} \sum_{j=1}^{n} \left(\boldsymbol{u}_j(\widehat{\boldsymbol{B}}_k)^\top \boldsymbol{u}_j(\widehat{\boldsymbol{B}}_k) \right). \end{aligned}$$

Define

$$d_k := \frac{\mathbb{E}\left(\boldsymbol{u}_j(\boldsymbol{B}_k)^\top \boldsymbol{u}_j(\boldsymbol{B}_k)\right)}{\mathbb{E}\left(\boldsymbol{y}_j^\top \boldsymbol{y}_j\right)}, \quad k = 1, 2, \ldots, \min(n, p).$$

We have the following results for the true and empirical reconstruction errors:

Theorem 4.1. *For any $k = 1, 2, \ldots, \min(n, p)$,*

$$P\left(\left|\left(R(\widehat{\boldsymbol{B}}_k) - R_n(\widehat{\boldsymbol{B}}_k)\right)\right| \leq d_k \left(\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right)\right) \sqrt{\frac{c\xi}{2n}}\right) \geq 1 - 2e^{-\xi},$$

and

$$P\left(0 \leq \left(R(\widehat{\boldsymbol{B}}_k) - R(\boldsymbol{B}_k)\right) \leq 2d_k \left(\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right)\right) \sqrt{\frac{c\xi}{2n}}\right) \geq 1 - 4e^{-\xi},$$

where c is a constant number.

In particular, as $\xi \to \infty$, if $d_k \left(\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right)\right) \sqrt{\frac{c\xi}{2n}} \longrightarrow 0$, then we have

$$\left|\left(R(\widehat{\boldsymbol{B}}_k) - R_n(\widehat{\boldsymbol{B}}_k)\right)\right| = O_p\left(d_k \left(\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right)\right) \sqrt{\frac{c\xi}{2n}}\right) = o_p(1). \tag{4.3.1}$$

In order to make the empirical reconstruction error close to the true reconstruction error with probability 1, we have two requirements:

- (1) ξ is large enough, which ensures $e^{-\xi} \to 0$ and the probability close to 1;
- (2) $d_k \left(\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right)\right) \sqrt{\frac{c\xi}{2n}} \to 0$ with k being fixed and $p, n \to \infty$.

For instance, $\xi = 10$ is large enough for the first requirement. We discuss more about the second requirement here. If $\mathbb{E}(y_{ij}^2)$ is finite, then $\left(\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right)\right) \sqrt{c\xi/2n}$ $= O(p/\sqrt{n})$. Therefore, to meet the second requirement, we need $d_k = o(\sqrt{n}/p)$. For example, if we assume $p/\sqrt{n} = O(1)$, which is common in high-dimensional statistics, then we just require $d_k = o(1)$. It is worth mention that, d_k should sat-

isfy this condition well with a fixed k if we assume a spiked covariance structure for the data.

We have mentioned that the transformed data always have finite variances, which indicates the proposed robust PCA method can achieve small excess error when the data has a spiked covariance structure. While under the same conditions, the standard PCA will fail to do so as the original data may have infinite variances, considering the exist of the outliers or heavy-tailed variables. In summary, the proposed method has robust statistical property, in the view of the excess error.

The rest of this section is the proof of Theorem 4.1. We discuss more about the spiked covariance structure in the next section (Section 4.3.2).

Proof 4.1. *We make use of the following lemma of the concentration inequality to complete our proof.*

Lemma 4.1. *(McDiarmid(1989)) Let $X_1, \ldots, X_n$ be n independent random variables taking values in $\mathcal{X}$ and let $Z = f(X_1, \ldots, X_n)$ where f is such that*

$$\sup_{x_1,\ldots,x_n,x_i^{'}\in\mathcal{X}} |f(x_1,\ldots,x_n) - f(x_1,\ldots,x_i^{'},\ldots,x_n)| \leq c_i,\ \forall\ 1 \leq i \leq n,$$

then

$$P[Z - \boldsymbol{E}(Z) \geq \xi] \leq e^{-2\xi^2/(c_1^2+\ldots+c_n^2)} \quad \textit{and} \quad P[\boldsymbol{E}(Z) - Z \geq \xi] \leq e^{-2\xi^2/(c_1^2+\ldots+c_n^2)}.$$

Let $\mathcal{X}$ be the set of all independent samples of $\boldsymbol{y}$ and

$$\begin{aligned} Z = f(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n) &= R(\widehat{\boldsymbol{B}}_k) - R_n(\widehat{\boldsymbol{B}}_k) \\ &= \boldsymbol{E}\left(\boldsymbol{u}(\widehat{\boldsymbol{B}}_k)^\top \boldsymbol{u}(\widehat{\boldsymbol{B}}_k)\right) - \frac{1}{n}\sum_{j=1}^{n}\left(\boldsymbol{u}_j(\widehat{\boldsymbol{B}}_k)^\top \boldsymbol{u}_j(\widehat{\boldsymbol{B}}_k)\right). \end{aligned}$$

Then we have $\forall 1 \leq i \leq n$,

$$\begin{aligned}
&\sup_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n,\boldsymbol{y}_i^{'}\in\mathcal{X}} \left| f(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n) - f(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_i^{'},\ldots,\boldsymbol{y}_n) \right| \\
&= \sup_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n,\boldsymbol{y}_i^{'}\in\mathcal{X}} \left| \frac{1}{n} \left(\boldsymbol{u}_i(\widehat{\boldsymbol{B}}_k)^\top \boldsymbol{u}_i(\widehat{\boldsymbol{B}}_k) - \boldsymbol{u}_i^{'}(\widehat{\boldsymbol{B}}_k)^\top \boldsymbol{u}_i^{'}(\widehat{\boldsymbol{B}}_k) \right) \right|.
\end{aligned}$$

Thus in order to apply Lemma 4.1, we only need to find the upper bound of the above quantity.

The following evaluation

$$\mathbb{E}\left(\boldsymbol{u}_j(\boldsymbol{B}_k)^\top \boldsymbol{u}_j(\boldsymbol{B}_k) \right) = \sum_{i=1}^{p} \mathbb{E}\left(u_{ij}^2\right) = O\left(d_k \sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right) \right),$$

which indicates

$$\boldsymbol{u}_j\left(\boldsymbol{B}_k\right)^\top \boldsymbol{u}_j\left(\boldsymbol{B}_k\right) = O_p\left(d_k \sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right) \right). \tag{4.3.2}$$

Then from (4.3.2), we have for any i *and* j,

$$\left| \frac{1}{n} \left(\boldsymbol{u}_i(\boldsymbol{B}_k)^\top \boldsymbol{u}_i(\boldsymbol{B}_k) - \boldsymbol{u}_j(\boldsymbol{B}_k)^\top \boldsymbol{u}_j(\boldsymbol{B}_k) \right) \right| = O_p\left(\frac{d_k}{n} \sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right) \right).$$

Let $c_j = \frac{cd_k}{n} \sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right)$ $(j = 1,\ldots,n)$ *with* c *being a constant which may be different from line to line. According to Lemma 4.1, we have*

$$\begin{aligned}
P(|Z - \boldsymbol{E}(Z)| \leq t) &\geq 1 - 2e^{-2t^2 / \left(\sum_{j=1}^{n} \left(\frac{cd_k}{n} \sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right) \right)^2 \right)} \\
&= 1 - 2e^{-2t^2 / \left(\frac{cd_k^2}{n} \left[\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right) \right]^2 \right)}.
\end{aligned}$$

Let $\xi = 2t^2 / \left(\frac{cd_k^2}{n} \left[\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right) \right]^2 \right)$, *which leads to* $t = d_k \left(\sum_{i=1}^{p} \mathbb{E}\left(y_{ij}^2\right) \right) \sqrt{\frac{c\xi}{2n}}$.

Then we can rewrite the above inequality as

$$P\left(\left|\left(R(\widehat{\boldsymbol{B}}_k)-R_n(\widehat{\boldsymbol{B}}_k)\right)\right|\leq d_k\left(\sum_{i=1}^{p}\mathbb{E}\left(y_{ij}^2\right)\right)\sqrt{\frac{c\xi}{2n}}\right)\geq 1-2e^{-\xi}, \tag{4.3.3}$$

which is the first part of Theorem 4.1.

For the second part of Theorem 4.1, we first have

$$R(\widehat{\boldsymbol{B}}_k)-R(\boldsymbol{B}_k)\geq 0 \quad \textit{and} \quad R_n(\widehat{\boldsymbol{B}}_k)-R_n(\boldsymbol{B}_k)\leq 0 \tag{4.3.4}$$

due to that $\boldsymbol{B}_k$ minimized the true reconstruction error and $\widehat{\boldsymbol{B}}_k$ minimized the empirical reconstruction error according to PCA. Hence we have

$$\begin{aligned}
0 &\leq R(\widehat{\boldsymbol{B}}_k)-R(\boldsymbol{B}_k) \quad (\textit{according to the first inequality in (4.3.4)})\\
&=\left(R(\widehat{\boldsymbol{B}}_k)-R_n(\widehat{\boldsymbol{B}}_k)\right)-\left(R(\boldsymbol{B}_k)-R_n(\boldsymbol{B}_k)\right)+\left(R_n(\widehat{\boldsymbol{B}}_k)-R_n(\boldsymbol{B}_k)\right)\\
&\leq\left(R(\widehat{\boldsymbol{B}}_k)-R_n(\widehat{\boldsymbol{B}}_k)\right)-\left(R(\boldsymbol{B}_k)-R_n(\boldsymbol{B}_k)\right)\\
&\quad (\textit{according to the second inequality in (4.3.4)})\\
&\leq\left|\left(R(\widehat{\boldsymbol{B}}_k)-R_n(\widehat{\boldsymbol{B}}_k)\right)\right|+\left|\left(R(\boldsymbol{B}_k)-R_n(\boldsymbol{B}_k)\right)\right|.
\end{aligned}$$

The first term is controlled by inequality (4.3.3). Following the same procedure, we also have

$$P\left(\left|R(\boldsymbol{B}_k)-R_n(\boldsymbol{B}_k)\right|\leq d_k\left(\sum_{i=1}^{p}\mathbb{E}\left(y_{ij}^2\right)\right)\sqrt{\frac{c\xi}{2n}}\right)\geq 1-2e^{-\xi}.$$

Therefore, with probability $1-4e^{-\xi}$

$$P\left(0\leq R(\widehat{\boldsymbol{B}}_k)-R(\boldsymbol{B}_k)\leq 2d_k\left(\sum_{i=1}^{p}\mathbb{E}\left(y_{ij}^2\right)\right)\sqrt{\frac{c\xi}{2n}}\right)\geq 1-4e^{-\xi}. \tag{4.3.5}$$

Inequality (4.3.3) and (4.3.5) are the final results in Theorem 4.1.

4.3.2 Behavior of the leading eigenvalues under spiked covariance structure

Data reconstruction from PCA is equivalent to estimation of the spiked structure for a covariance matrix. A variety of literature have made effort to understand the behaviour of the empirical eigenvalues under the high dimensional setting when the sample size n and the dimension p both go to infinity, see for example, Johnstone [2001]; Lam et al. [2011]; Lee et al. [2014]; Cai et al. [2017]. Particularly, we are interested in the spiked covariance model, of which the distribution of the empirical eigenvalues has been studied in Wang and Fan [2017], Cai et al. [2017], and others. In the aspect of the empirical eigenvalues, how does the PCA benefit from our proposed method? Besides, how does the transformation in our method affect the spike covariance structure? We make use of some simulations and the conclusions from the literature to answer the above questions.

Remark 4.3. *The spiked covariance model typically assumes that there are several eigenvalues larger than the rest. The larger eigenvalues are called the spiked eigenvalues, and the remaining ones are called the non-spiked eigenvalues. Specifically, Wang and Fan [2017] and Cai et al. [2017] assume that the population covariance matrix has k ($k/p \to 0$, p is the number of dimension) well separated spiked eigenvalues and the non-spiked eigenvalues are all bounded but otherwise arbitrary.*

In Wang and Fan [2017], the asymptotic normality of the spiked empirical eigenvalues was proved under a spiked covariance model (see Assumption 2.1 to 2.3 and Theorem 3.1 in Wang and Fan [2017]). Assume the population covariance model has k spiked eigenvalues $\{\lambda_j\}_{j=1}^k$, and the corresponding empirical eigenvalues are $\{\widehat{\lambda}_j\}_{j=1}^k$. The theorem shows that $\widehat{\lambda}_j/\lambda_j (j = 1, 2, \ldots, k)$ are asymptotic normal and the bias of $\widehat{\lambda}_j/\lambda_j$ is controlled by a term of rate $c_j = p/(n\lambda_j)$, where n is the sample size and p is the dimension. To make $\widehat{\lambda}_j$ asymptoticly unbiased, it requires $c_j \to 0$ for $j \leq k$. In our proposed method, we do a data

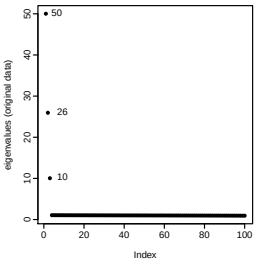
transformation first, and then conduct the classical PCA on the transformed data. Therefore, given the same spiked covariance model on the transformed data, the asymptotic normality proved in Wang and Fan [2017] generally holds for our method. When the original data does not have a finite population covariance or it can not satisfy the assumptions in Wang and Fan [2017], the standard PCA is not valid on the data. On the other hand, the transformed data in our method always has finite population covariance and those assumptions are satisfied, which solves the problem of the standard PCA.

Note that, the unbiased estimation depends on $c_j = p/(n\lambda_j)$. That is, with the same p and n, a larger λ_j can result in smaller bias. Then an interesting problem is, if the original data has a spiked covariance structure, how does the transformation change the spiked population eigenvalues and hence change the behaviour of the empirical spiked eigenvalues? We use some simulations to illustrate the effect. We have two examples, with Example 1 for normal distributed data and Example 2 for heavy-tail distributed data. The data generating process is as follows.

We simulate $P \times 1$ vector $\boldsymbol{y}_n$ $(n = 1, 2, \ldots, N)$ by

$$\boldsymbol{y}_n = \sum_{i=1}^{3} \alpha_i \boldsymbol{b}_i k_{i,n} + \boldsymbol{\varepsilon}_n, \tag{4.3.6}$$

where $\boldsymbol{b}_i$ $(i = 1, 2, 3)$ are $P \times 1$ vectors generated by a QR decomposition, $k_{i,n}$ $(i = 1, 2, 3,\ n = 1, 2, \ldots, N)$ are independently generated from standard normal $N(0, 1)$ for Example 1 and from t-distribution with degree of freedom 2 for Example 2, and $\boldsymbol{\varepsilon}_n$ $(n = 1, 2, \ldots, N)$ are the $P \times 1$ error vectors with elements independently generated from $N(0, 1)$. Besides, $(\alpha_1, \alpha_2, \alpha_3) = (7, 5, 3)$. For all the cases, $P = 100$. We intend to compare the spiked population eigenvalues (the first three eigenvalue according to the data generating process) of the original data and the transformed data, as well as the empirical spiked eigenvalues with different sample sizes.



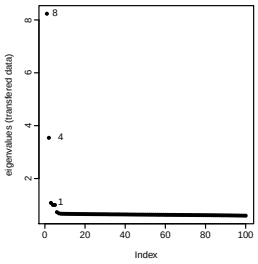
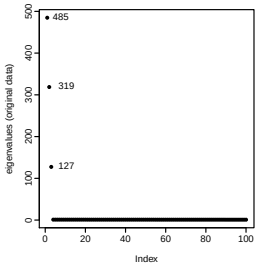


Figure 4.1: Approximated population eigenvalues for Example 1: normal distribution ($N(0,1)$)



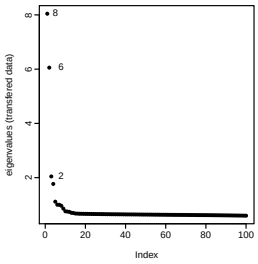


Figure 4.2: Approximated population eigenvalues for Example 2: heavy-tail distribution ($t(2)$)

In order to approximate the population eigenvalues, we let $N = 100000$, and compute the eigenvalues of the sample covariance matrix of the original data and the transformed data, respectively. The approximated population eigenvalues for both examples are shown in Figure 4.1 and 4.2. We see for both examples, the original data has three spiked eigenvalues, while the transformed data has two larger eigenvalues and the third one is mixed with the non-spiked part. This is not a surprise as a transformation on the original data may cause the loss of some information. Nevertheless, the transformed data still has well separated spiked eigenvalues, and the standard PCA is still valid on the data. Furthermore, comparing Example 1 and Example 2, we can notice that the heavy-tailed data produces much larger original spiked eigenvalues, while the transformed spiked eigenvalues look similar.

Further, we simulate data with different sample sizes ($N = (50, 100, 500, 1000, 5000)$, each with 1000 replicates) to have a look at the effect of the transformation on the behaviour of the empirical spiked eigenvalues. Note that $N = 50$ and $N = 100$ provide the high dimensional cases and the rests give large sample cases. The biases and standard deviation of the largest three spiked eigenvalues (eg. the average and standard deviation of $\widehat{\lambda}_i/\lambda_i - 1,\ i = 1, 2, 3$ which are the largest 3 eigenvalues) for both Examples are shown in Table 4.1, 4.2, and 4.3.

Firstly, comparing the three tables we find rpca performs the worst in esti-

Table 4.1: Bias and SD of $\widehat{\lambda}_1/\lambda_1 - 1$

		50		100		500		1000		5000	
		Bias	SD	Bias	SD	Bias	SD	Bias	SD	Bias	SD
normal	cpca	**0.051**	0.195	**0.030**	0.137	**0.005**	**0.062**	**0.004**	**0.044**	**−0.001**	**0.020**
	rpca	0.114	**0.120**	0.033	**0.092**	−0.034	0.066	−0.046	0.060	−0.052	0.054
t(2)	cpca	0.518	7.281	1.366	38.055	3.973	86.691	0.713	8.141	1.089	11.464
	rpca	**0.180**	**0.124**	**0.063**	**0.099**	**−0.016**	**0.073**	**−0.028**	**0.068**	**−0.036**	**0.065**

mating λ_3 (Table 4.3), especially compared with cpca in the example of normal distribution. It is not surprising as the population λ_3 is mixed with the non-spiked part and no longer a spiked eigenvalue (see Figure 4.1 and 4.2). In addition, according to the rate $c_j = P/(N\lambda_j)$, it needs a very large N to make the estimation of the third one unbiased if λ_j is small.

Table 4.2: Bias and SD of $\widehat{\lambda}_2/\lambda_2 - 1$

		50		100		500		1000		5000	
		Bias	SD	Bias	SD	Bias	SD	Bias	SD	Bias	SD
normal	cpca	**0.034**	0.184	**0.018**	0.142	**0.004**	**0.063**	**0.005**	**0.043**	**0.002**	**0.020**
	rpca	0.341	**0.159**	0.138	**0.124**	−0.023	0.074	−0.042	0.066	−0.062	0.056
t(2)	cpca	−0.584	0.475	−0.523	0.276	−0.342	0.450	−0.285	0.551	−0.162	0.399
	rpca	**0.098**	**0.131**	**0.001**	**0.109**	**−0.086**	**0.081**	**−0.092**	**0.075**	**−0.101**	**0.070**

Secondly, apart from the information lose of the third spiked eigenvalue, from Table 4.1 and 4.2, we see rpca has advantage in estimating the two spiked eigenvalues, especially in the heavy-tailed example. For Example 1 (the normal case), although rpca has relatively larger bias but it has smaller variance. In total rpca is not worse than cpca when the data is normal-distributed. More importantly, we see rpca provides much less biased and more stable estimations for Example 2 (the heavy-tail case) comparing to cpca. Hence the asymptotic normal results are still valid on the transformed heavy-tailed data but not suitable for the original data. It provides a strong evidence that the classical PCA is not valid under heavy-tailed data while our proposed robust PCA works well in such situations.

Table 4.3: Bias and SD of $\widehat{\lambda}_3/\lambda_3 - 1$

		50		100		500		1000		5000	
		Bias	SD	Bias	SD	Bias	SD	Bias	SD	Bias	SD
normal	cpca	**0.140**	0.189	**0.062**	0.134	**0.013**	0.064	**0.001**	**0.044**	**−0.003**	**0.020**
	rpca	2.137	**0.117**	1.248	**0.073**	0.331	**0.058**	0.193	0.063	0.077	0.065
t(2)	cpca	**−0.629**	0.240	−0.562	0.274	−0.424	0.358	−0.360	0.329	−0.226	0.402
	rpca	0.960	**0.160**	**0.510**	**0.129**	**0.111**	**0.087**	**0.062**	**0.077**	**0.019**	**0.064**

4.4 Reconsturction performance under different situations

In this section, we illustrate the advantage of our proposed method (rpca) against the classic PCA (cpca) in recovery of orginal data under several scenarios. Throughout the simulations, we use mean squared error (MSE) of the approximation to measure the performance:

$$\text{MSE} = \sum_{n=1}^{N} \|\widehat{\boldsymbol{y}}_n - \boldsymbol{y}_n\|_2^2/(NP),$$

where the $\widehat{\boldsymbol{y}}_n$ is the approximation from rpca or cpca (both recovering at least 80% of the total variance) and $\boldsymbol{y}_n$ is the original data. N is the sample size and P is the dimension of $\boldsymbol{y}_n$.

Example 1 shows the powerful ability of rpca to handle data with heterogeneity in variances. Example 2 demonstrates that rpca performs better than cpca when approximating data with outliers. Further, in Example 3, we simulate data from three different heave-tailed distributions, as well as the normal distribution as a benchmark, and we find that the rpca can recover those data more accurately than the cpca. Now let us discuss the simulations in details.

4.4.1 Example 1 : heterogeneity in variances

The heterogeneity in variances is ubiquitous in real-life data, and the variables with extreme significant variances tend to dominate the results of classic PCA (Jolliffe [2002]). Hence, the information contained in other variables is masked, which makes the classic PCA less informative. In this example, we show that rpca can deal with this problem and recover the original data more precisely.

We simulate $\boldsymbol{y}_n : P \times 1$ by $(\boldsymbol{y}_n^{(1)\top}, \boldsymbol{y}_n^{(2)\top})^\top (n = 1, 2, \ldots, N)$, where $\boldsymbol{y}_n^{(1)}$ and $\boldsymbol{y}_n^{(2)}$ are $(P/2) \times 1$ vectors generated by

$$\boldsymbol{y}_n^{(1)} = \sum_{i=1}^{3} \alpha_i \boldsymbol{b}_i^{(1)} k_{i,n}^{(1)} + \boldsymbol{\varepsilon}_n^{(1)}, \quad \boldsymbol{y}_n^{(2)} = \sum_{i=1}^{3} \alpha_i \boldsymbol{b}_i^{(2)} k_{i,n}^{(2)} + \boldsymbol{\varepsilon}_n^{(2)}$$

where $\boldsymbol{b}_i^{(1)}$ and $\boldsymbol{b}_i^{(2)}$ $(i = 1, 2, 3)$ are $(P/2) \times 1$ vectors independently generated by two QR decompositions. $k_{i,n}^{(1)} (i = 1, 2, 3,\ n = 1, 2, \ldots, N)$, are independently generated from $N(0, 1)$ while $k_{i,n}^{(2)} (i = 1, 2, 3,\ n = 1, 2, \ldots, N)$ are those from $N(0, 0.1)$. $\boldsymbol{\varepsilon}_n^{(1)}$ and $\boldsymbol{\varepsilon}_n^{(2)}$ are both the $(P/2) \times 1$ error vectors with elements independently generated from $N(0, 1)$. Besides, $(\alpha_1, \alpha_2, \alpha_3) = (7, 5, 3)$.

Thus, $\boldsymbol{y}_n$ consists of two parts with widely different variances. We can visualize the data and variance of a 100×100 sample matrix of $\boldsymbol{y}_n$ in Figure 4.3 and 4.4. In both figures, the colour represents the size of the value: the darker the colour, the larger the value. Figure 4.3 shows the original data matrix, and we can see clearly that some of the left parts have much more variations than the rest. The top part of Figure 4.4, which shows the sample variances of the original data, displays the widely differing variances more clearly. However, from the bottom part of Figure 4.4, which shows the sample variances of $e^{\mathrm{i}y_{in}} (i = 1, 2, \ldots, P)$, we see the differences in the variances are decreased after transforming the data. The transformation helps reduce the effect of the heterogeneity in variances on the results of PCA.

Next we compare the performance of cpca and rpca on approximating the

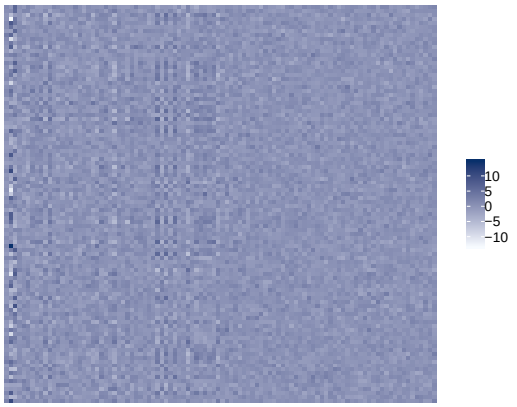


Figure 4.3: example 1, data

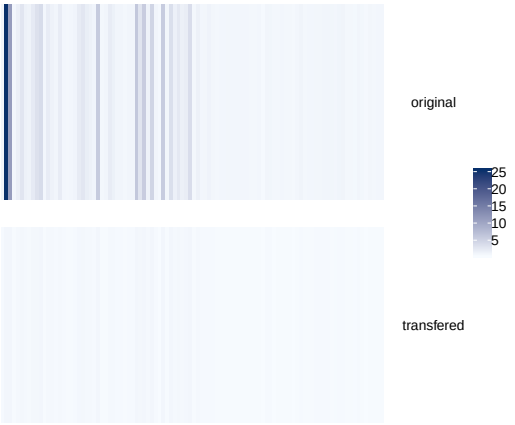


Figure 4.4: example 1, variance

data. We simulate this example for different sets of (P, N) : $(50, 40), (50, 100)$, $(100, 100), (100, 200), (200, 190)$, which includes the situations of $P < N, P = N$ as well as $P > N$. Besides, although the value of P and N are not extremely large, we can consider $(50, 40), (100, 100)$ and $(200, 190)$ as high dimensional settings because the ratios $P/N \geq 1$. The average MSEs for 1000 simulations are shown in Table 4.4. It is clear that rpca performs better than cpca on recovering data with widely differing variances. For such data, classic PCA focus on those variables with large variances but ignores others which may be also very important. However, our proposed method automatically shrinks those differences, which is shown in Figure 4.4, therefore results in a more accurate approximation.

Table 4.4: average MSE, 1000 simulations, Example 1

(P, N)	(50, 40)	(50, 100)	(100, 100)	(100, 200)	(200, 190)
rpca	**0.203**	**0.211**	**0.201**	**0.204**	**0.193**
cpca	0.498	0.513	0.357	0.360	0.279

4.4.2 Example 2: outliers

As the volume of data increasing, it is common to have outliers in the data. This example simulates data with outliers and shows that our proposed method is robust to such kind of data since the transformation can decrease the extreme of outliers.

We first simulate $P \times 1$ vector $\boldsymbol{y}_n$ by

$$\boldsymbol{y}_n = \sum_{i=1}^{3} \alpha_i \boldsymbol{b}_i k_{i,n} + \boldsymbol{\varepsilon}_n$$

which is exactly the same as how we generated $\boldsymbol{y}_n^{(1)}$ in Example 1 except with dimension P instead of $P/2$. After simulating N samples, we have a matrix $\boldsymbol{Y} : P \times N$, whose columns consist of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$. Then we randomly replace 14.4% of the elements in this matrix with values independently generated from $N(0, 36)$. Thus about 14.4% of the elements in $\boldsymbol{Y}$ are outliers.

The same as Example 1, we show values and variances of a 100×100 sample for Example 2 in Figure 4.5 and 4.6. We can see clearly some squares with extremely darker or lighter colour than the others in Figure 4.5, and those are outliers. From Figure 4.6, we see there are some huge variances (top part of the figure) in the original data caused by the outliers, which is not a good sign for standard PCA, while our method can shrink those differences (bottom part of the figure) by the proposed transformation. We try different sets of (P, N) (which are the same as Example 1) and report the average MSEs of 1000 simulations in Table 4.5. It is not surprising that rpca performs better than cpca, as rpca cuts back the differences between the average values and the outliers.

Table 4.5: average MSE, 1000 simulations, Example 2

(P, N)	(50, 40)	(50, 100)	(100, 100)	(100, 200)	(200, 190)
rpca	**0.225**	**0.234**	**0.215**	**0.218**	**0.201**
cpca	0.586	0.603	0.451	0.455	0.377

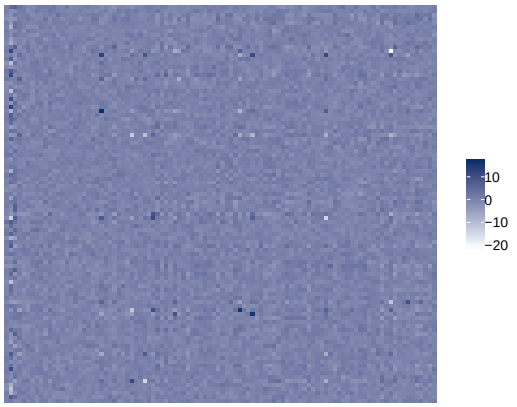


Figure 4.5: example 2, data

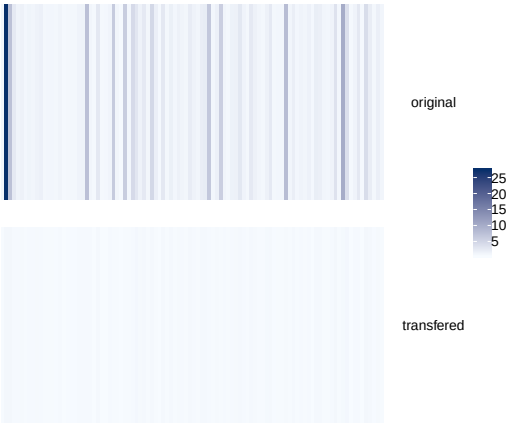


Figure 4.6: example 2, variance

4.4.3 Example 3: heavy-tailed data

Now we consider if rpca works well for data from different kinds of heavy-tailed distributions. There are a large amount of real-world data which have been proved to be heavy-tailed, therefore it is highly possible that a dataset with large dimensions contains heavy-tailed variables. We simulate data from t distribution, Pareto distribution and Cauchy distribution, which are all very common heavy-tailed distributions in real-world data. As a benchmark, we also simulate data from the normal distribution.

We simulate $P \times 1$ vector $\boldsymbol{y}_n$ by

$$\boldsymbol{y}_n = \sum_{i=1}^{3} \alpha_i \boldsymbol{b}_i k_{i,n} + \boldsymbol{\varepsilon}_n$$

which is the same as the first step in Example 2, except $k_{i,n}$ $(i = 1, 2, 3,\ n = 1, 2, \ldots, N)$, are independently generated from $N(0, 1)$ for the normal distribution, $t(2)$ for the t distribution, $pareto$ $(scale = 0.5, shape = 1.5)$ for the Pareto distribution (by function 'rpareto' in R package 'VGAM'), and $cauchy$ $(location = 0, scale = 1)$ for the Cauchy distribution. For this example, we try $(P, N) = (100, 100), (200, 190)$ and the average MSEs of 1000 simulations are shown in

Table 4.6.

Firstly, we see that on the normal-distributed data, the performance of rpca is better than that of cpca while the differences are not extremely large, which means on the normal-distributed data our proposed method is at least not worse than the standard PCA. Secondly, for the data from the three heavy-tailed distributions, rpca performs much better than cpca. One of the reasons for the worse performance of cpca is the uncertainty of the second moments of the heavy-tailed data. For example, the Cauchy distribution has no finite second moments, which makes the sample covariances estimated in cpca invalid and leads to the extremely bad performance shown in Table 4.6. However, the transformation of rpca guarantees that the transformed data has finite second moments, which ensures the feasibility of PCA on transformed data.

Table 4.6: average MSE, 1000 simulations, Example 3

(P, N)	(100, 100)				(200, 190)			
	Normal	t	Pareto	Cauchy	Normal	t	Pareto	Cauchy
rpca	**0.212**	**0.235**	**0.200**	**0.258**	**0.198**	**0.221**	**0.194**	**0.245**
cpca	0.335	1.174	1.318	127.242	0.278	0.840	1.058	388.595

4.5 Empirical application

In this section, we performed the robust PCA on a real dataset to demonstrate an example of applying the method in real data analysis. The data, which has 77 variables and 1080 samples, comes from Higuera et al. [2015], in which the details of the experiment and the measurements can be found. The data consists of the protein expression measurements of 77 proteins obtained from normal genotype control mice and Down syndrome (DS) mice, both with and without shock and drug treatments. There were 72 mice in the experiment, and 15 measurements of each protein per mouse were recorded. Thus there are 1080 (=72x15) expression measurements for each protein. We did a preprocessing step to deal with missing

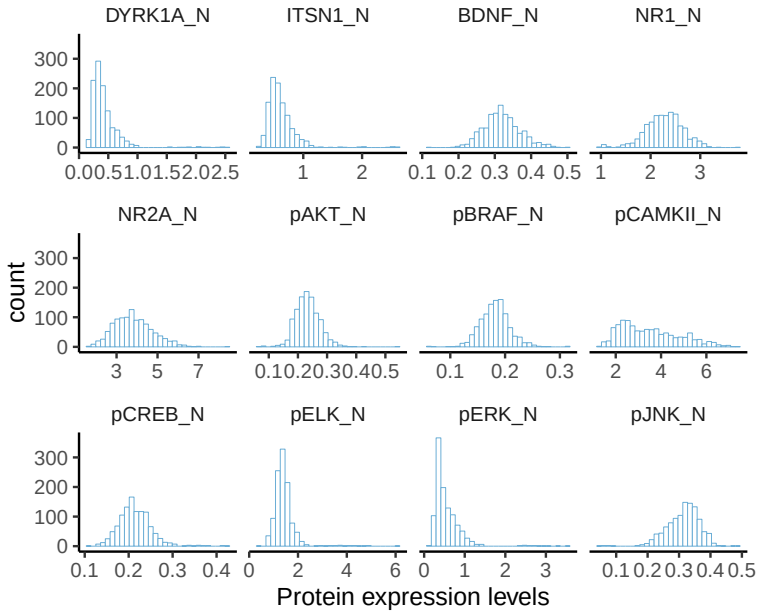


Figure 4.7: The histogram of the expression measurements for the first 12 proteins

values.

Figure 4.7 shows the histograms of the expression measurements for the first 12 proteins in the data. We can see that although some proteins have nearly normal distributed expression levels, most of the proteins, such as DTRK1A, ITSN1, pCAMKII, and pERK, have heavy tails or extreme outliers in their expression levels. Thus, it is reasonable to statistically analyse this data with robust methods.

We first compare the approximations from robust PCA (rpca) and classic PCA (cpca) for the whole dataset under four criteria: the mean squared error (MSE) of the low rank representation, the number of principals we extracted based on threshold 0.8 of the total variance, the estimated smallest spiked eigenvalue, as well as the spiked ratio $P/(N\widehat{\lambda}_{\widehat{r}})$ (which we discussed in Section 4.3.2), which are shown in Table 4.7. Both of the spiked ratios are small, with the rpca one larger than that of cpca. It could indicate that rpca reduces the spiked eigenvalues and the smallest spiked one is more biased than that of cpca. On the other hand, with seven numbers of eigenvalues selected under threshold 0.8,

rpca reaches a better approximation performance than cpca. It is worth mentioned that, although rpca is more flexible due to larger $\widehat{r}$ than cpca, the better out-of-sample performance provided later illustrates its appropriate flexibility.

Table 4.7: The comparison of rpca and cpca on the whole data

	MSE	$\widehat{r}$	ratio $(\frac{P}{N\widehat{\lambda}_{\widehat{r}}})$	$\widehat{\lambda}_{\widehat{r}}$
rpca	**0.009**	7	0.558	0.126
cpca	0.014	3	0.137	0.512

One potential analysis for this dataset is using the protein expression levels to classify the mice. There were 38 control mice and 34 DS mice. The experiment in Higuera et al. [2015] involved shock and drug treatment for the treatment and control groups. The shock treatment consisted of two types, one was context-shock (CS), which allowed the mice to explore a novel cage for several minutes and then gave a brief electric shock, and the other one was shock-context (SC), which did the inverse. Including the with and without the drug memantine, the mice are separated into eight groups. Hence, each group has 7 to 9 mice. Table 4.8 shows the number of mice in each class. "c" represents the control group and "t" is the test group, which consists of DS mice. "m" represents the drug memantine and "s" is saline, which performs as a placebo.

Table 4.8: Number of mice in each class, from Higuera et al. [2015]

	Classes	No. of mice
Control mice	c-SC-s	9
	c-SC-m	10
	c-CS-s	9
	c-CS-m	10
Down syndrome(DS) mice	t-SC-s	9
	t-SC-s	9
	t-SC-s	7
	t-SC-s	9

We conduct a classification with a subset of the data for groups "c-CS-s" and "t-CS-s" by using the principal logistic regression. For these two groups, the

shock and drug treatment were the same, but the genotype is different. One group consists of the normal mice while the other group consists of the DS mice. By Higuera et al. [2015], the comparison of these two groups is biologically meaningful as it is related to the initial trisomy vs. control differences. We aim to use the protein expression levels through principal logistic regression to identify DS mice from the normal ones.

The subset has 240 measurements and 77 proteins. We first split the data into training (75%) and test sets (25%) by random, in order to measure the prediction performance of the rpca and cpca by the cross-validation. Then we apply the rpca and cpca on the training data, extract the eigenvectors and construct the principal components as the design matrix for the logistic regression. For rpca, the principal design matrix is constructed by $\widehat{\boldsymbol{B}}^\top[\boldsymbol{Y}^\top, \boldsymbol{Y}^\top]^\top$, where $\widehat{\boldsymbol{B}}$ is the $(2P) \times \hat{r}_1$ eigenvector matrix of the transformed training data, $\hat{r}_1$ is the estimated number of eigenvalues of rpca, and $\boldsymbol{Y}$ is the original training data with P variables. For cpca, the corresponding principal design matrix is $\widehat{\boldsymbol{D}}^\top \boldsymbol{Y}$, where $\widehat{\boldsymbol{D}}$ is the $P \times \hat{r}_2$ eigenvector matrix of the original data, $\hat{r}_2$ is the estimated number of eigenvalues of cpca, and $\boldsymbol{Y}$ is the original data. Then, we use the principal design matrices as well as the class labels to fit logistic models and compute the prediction values for the test set. If the prediction value is larger than 0.5, we set it to be "t-CS-s", otherwise "c-CS-s". At last, we record the prediction accuracies for both of the methods. We repeat the process for 1000 times to ensure we have different training and test sets. Figure 4.8 shows the histogram of the prediction accuracies and the mean accuracy for both methods. We can see that when using the principal design matrix constructed from robust PCA to fit the logistic model, almost all the prediction accuracy are larger than 0.5 and most of them are around 0.78. However, cpca performs much worse than rpca, with most of the prediction accuracy near 0.68. This is because heavy-tailed measurements and outliers affect the validity of the cpca, while the rpca method reduces those effects and results in a better performance. Our proposed method can help iden-

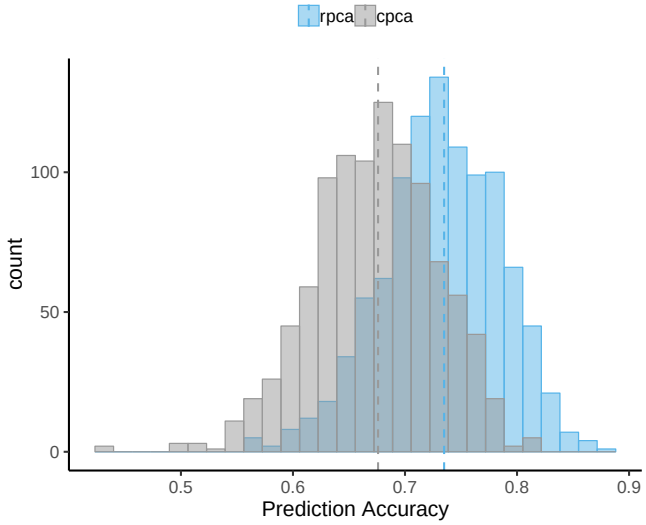


Figure 4.8: Comparing the classification on mice data

tify DS mice from the normal ones by the protein expression levels effectively. This example shows that the robust PCA can definitely perform an essential role in classification models and also other statistic analysis.

4.6 Conclusion

In this chapter, we addressed the challenge of applying the PCA on the high-dimensional data in the presence of various kinds of heterogeneities. Specifically, we proposed a robust PCA, based on a characteristic-function-type of transformation, to deal with the potential heterogeneities, which is particularly useful when the data is heavy-tailed (for example, with infinite variance). We show that the method is more robust than the classical PCA in the view of the excess error, assuming a spiked covariance structure for the data. We also studied the impact of the transformation on the spikeness of the spiked covariance structure. We illustrate with simulations that the transformation shrinks the spikeness while still keeps a well separable spiked covariance matrix. Particularly, the proposed method should work well when the original data has infinite variance, while the

classical method is invalid. Simulations and empirical analysis show that the proposed robust PCA method is better than the classical PCA method, with the exist of heterogeneities in the data. As a by product, the proposed method is able to detect the non-linear relationships between the variables.

Chapter 5

Conclusion and Future Work

This thesis considered topics of modelling and forecasting high-dimensional data with methods related to the Principal Component Analysis (PCA). In Chapter 2 and 3, we studied data-adaptive methods to improve the fitting and forecasting performance of mortality data, which is a representative high-dimensional time series data. In Chapter 4, we proposed a novel robust version of PCA to deal with heterogeneities in high-dimensional data.

We contributed to seeking linear features to attain optimal forecasting of the US mortality data in Chapter 2 and proposed a two-style factor model with two types of features. Both the two kinds of features were proved to enjoy equally fast rates of convergence. The method did improve the forecasting performance in our empirical and simulating analysis. In Chapter 3, we applied a time-varying factor model to fit the mortality data, in order to allow time-varying factor loadings (the relationship between age variables and the Mortality Index) in mortality modelling. Accordingly, we proposed two forecasting methods for the time-varying factor loadings and studied the optimal "boundary" between the short-term and long-term forecasting, which was favored by the two forecasting methods, respectively. Simulations and empirical studies showed the proposed method was able to recover the underlying time-varying factor loadings, which also resulted in good forecasting performance. The work were motivated by the mortality data, but they are widely applicable to high-dimensional time series data in other areas, which have similar structure to the mortality data.

Finally, in Chapter 4, a robust version of PCA was proposed. We constructed a transformation, motivated by the characteristic function, to improve the robustness of the standard, or classical, PCA. We showed the method was more robust than the classical PCA, in the sense of its ability to achieve accurate estimation of the reconstruction error even with extremely heavy-tailed data. The application on the mice data showed that making use of the proposed robust PCA was able to improve the classification accuracy of different types of mice.

The work presented in this thesis leaves several interesting topics for further research. For instance, we can extend the two-style factor model in Chapter 2 to jointly forecast mortality data in multiple countries by combining it with grouped forecasting method (such as the optimal-combination method proposed in Hyndman et al. [2011]). In literature, Li and Lee [2005]; Enchev et al. [2017]; Shang and Haberman [2020] and many others modeled the mortality data in similar countries as groups, and showed that jointly modelling improved the forecasting for individual country significantly. But most of the multi-population models were only based on the Lee-Carter model. As we have shown the more accurate forecasting of our method comparing to the Lee-Carter model in the single-population modelling, it is expected that the two-style model can also have better performance in the framework of jointly forecasting. Therefore, it is interesting to extend our two-style factor model from forecasting a single country individually to forecasting multiple countries jointly. Moreover, the statistical properties of the new multi-population two-style factor model is worthwhile to be studied.

Another potential work is related to the robust PCA in Chapter 4. The proposed robust PCA can be seen as a kernel PCA method. While we did not study from this aspect in details, we can do further research for the robust PCA under the framework of kernel PCA and explore its potential extensions. For example, we may study the theoretical behavior of the eigen-space produced by our method following the techniques in Reiß and Wahl [2020]. In Reiß and Wahl

[2020], the authors provided non-asymptotic upper bounds for the excess risk of the reconstruction error and discussed how these results could be transferred to the subspace distance (the distance between the subspace spanned by the empirical eigenvectors and the one spanned by the population version). More importantly, they allowed for general Hilbert spaces to include the kernel PCA. Therefore, it is worthwhile to establish the subspace distance results of the robust PCA in Chapter 4 under the framework of kernel PCA.

Appendix of Chapter 2

A.1 Additional simulations

In this appendix, we provide more simulation studies as a supplementary to Section 2.5 in Chapter 2. Specially, *Example* 5 and 6 are special cases to the examples in Section 2.5, in which the two different types of features are completely separated.

Firstly, in *Example* 4 and 5, we show that the first step of our method extracts features with strong time-serial dependence, which indicates a powerful forecasting ability. In addition, the low-dimensional representation has relative small reconstruction errors, which is necessary for recovering the data.

Secondly, with *Example* 5 and 6, we show that our method performs better on forecasting compared to static PCA and dynamic PCA.

For descriptive convenience, we use "SWPCA" to represent our method, "CPCA" to represent the static PCA which was described in Section 2.3, and "DPCA" to represent the dynamic PCA described in Section 2.3 with $\ell_0 = 1$. We also consider comparing with the method given in Lam et al. [2011], and we use "DPCA(ℓ)" to represent it, with $\ell = 1, 5, 10$. "DPCA(1)" is the same with the first step of our method, while "DPCA(5)" and "DPCA(10)" aggregate more auto-covariances but discarding the variance.

Data generating

- *Example 4*

  $\{\boldsymbol{y}_t\}_{t=1,2,...,T} : P \times 1$ is generated by

  $$\boldsymbol{y}_t = \boldsymbol{a} + \boldsymbol{b}k_t + \boldsymbol{\varepsilon}_t,$$

  where $\boldsymbol{a}$ is a $P \times 1$ mean vector with elements generated from standard normal, $\boldsymbol{b}$ is a $P \times 1$ vector obtained by the first column of a QR decomposition of a random generated matrix, $\{k_t\}_{t=1,2,...,T}$ is generated from $AR(1)$ model with coefficient 0.7 and mean 0, and $\boldsymbol{\varepsilon}_t$ is a $P \times 1$ error term with elements generated from standard normal.

- *Example 5*

  Construct $\boldsymbol{y}_t = (\boldsymbol{y}_t^{(1)\top}, \boldsymbol{y}_t^{(2)\top})^\top$. $\{\boldsymbol{y}_t^{(1)}\}_{t=1,2,...,T} : (dP) \times 1$ is generated by

  $$\boldsymbol{y}_t^{(1)} = \boldsymbol{b}k_t + \boldsymbol{\varepsilon}_t^{(1)},$$

  where $\boldsymbol{b}$ is a $(dP) \times 1$ vector with elements generated from $U(0,1)$, $\{k_t\}_{t=1,2,...,T}$ is generated from $AR(1)$ model with coefficient 0.8, and $\boldsymbol{\varepsilon}_t^{(1)}$ is a $(dP) \times 1$ error term with elements independently generated from $N(0, 0.2)$.

  $\{\boldsymbol{y}_t^{(2)}\}_{t=1,2,...,T} : ((1-d)P) \times 1$ is generated by

  $$\boldsymbol{y}_t^{(2)} = \boldsymbol{a}w_t + \boldsymbol{\varepsilon}_t^{(2)},$$

  where $\boldsymbol{a}$ is a $((1-d)P) \times 1$ vector with elements generated from $U(0,1)$, $\{w_t\}_{t=1,2,...,T}$ is generated from $N(0, 1.5)$, and $\boldsymbol{\varepsilon}_t^{(2)}$ is a $((1-d)P) \times 1$ error term with elements independently generated from $N(0, 0.2)$.

  We call $\{\boldsymbol{y}_t^{(1)}\}_{t=1,2,...,T}$ the dependent part as it has autocorrelations within observations (time dimension), and $\{\boldsymbol{y}_t^{(2)}\}_{t=1,2,...,T}$ the independent part as it has independent generated observations, while the variance of it is

larger than that of $\{\boldsymbol{y}_t^{(1)}\}_{t=1,2,\ldots,T}$. The parameter d is the proportion of the dependent part among the whole dataset, with possible values among $(0,1)$. As a result of this design, the whole data consists of two part. The dependent part has strong serial dependence with relatively small variance and the independent part has very weak dependence with relatively large variance. This is a special case for the two-style factor model in example 1 to 3 in which we can set 0s in the coefficient vector $\boldsymbol{a}$ and $\boldsymbol{b}$ to get the example 5.

- *Example 6*

  The data structure is the same with *Example* 5 with $d = 0.4$, except that $\{k_t\}_{t=1,2,\ldots,T}$ is generated from $AR(1)$ model with coefficient 0.7, $\boldsymbol{\varepsilon}_t^{(1)}$ and $\boldsymbol{\varepsilon}_t^{(2)}$ are $(dP) \times 1$ error terms with elements independently generated from $N(0, 0.5)$, and $\{w_t\}_{t=1,2,\ldots,T}$ is generated from $N(0, 3)$. The main difference is we enlarge the variations in *Example* 3, comparing with *Example* 6. The purpose is to show that keeping sufficient information of the variation is necessary.

Because in *Example* 5 and 6, $\boldsymbol{y}_t$ consists of the dependent part and independent part, we also report the FRMSE for the two parts separately, in addition to the overall FRMSE defined in Section 2.5. Rewrite $\widehat{\boldsymbol{y}}_{T-i}$ as $(\widehat{\boldsymbol{y}}_{T-i}^{(1)\top}, \widehat{\boldsymbol{y}}_{T-i}^{(2)\top})^\top$ and $\boldsymbol{y}_{T-i}$ as $(\boldsymbol{y}_{T-i}^{(1)\top}, \boldsymbol{y}_{T-i}^{(2)\top})^\top$, then:

$$\text{Dependent FRMSE}(h) = \left(\frac{\sum_{i=0}^{h-1} \|\widehat{\boldsymbol{y}}_{T-i}^{(1)} - \boldsymbol{y}_{T-i}^{(1)}\|_2^2}{hPd} \right)^{1/2},$$

$$\text{Independent FRMSE}(h) = \left(\frac{\sum_{i=0}^{h-1} \|\widehat{\boldsymbol{y}}_{T-i}^{(2)} - \boldsymbol{y}_{T-i}^{(2)}\|_2^2}{hP(1-d)} \right)^{1/2},$$

where d is the proportion of $\boldsymbol{y}_t^{(1)}$ among $\boldsymbol{y}_t$.

Results

We try different sets of (P, T): $(50, 50)$, $(50, 100)$, $(100, 100)$, $(100, 200)$, $(200, 200)$, as we would like to evaluate the performance under situations P and T are comparable. The results are shown in Table A.1 to Table A.6.

Table A.1: Variance and Dependence of $\widehat{k}_t$

	Time variance ($\widehat{k}_t$)			Time dependence ($\widehat{k}_t$)			Mix ($\widehat{k}_t$)		
(P, T)	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA
					Example 4				
$(50, 50)$	**7.882**	7.723	6.073	1.201	1.607	**1.833**	9.083	**9.331**	7.905
$(50, 100)$	**6.001**	5.917	4.595	1.061	1.328	**1.455**	7.062	**7.245**	6.050
$(100, 100)$	**7.968**	7.811	6.094	0.957	1.392	**1.671**	8.925	**9.204**	7.765
$(100, 200)$	**5.996**	5.911	4.567	0.944	1.241	**1.409**	6.940	**7.151**	5.976
$(200, 200)$	**7.974**	7.814	6.088	0.757	1.130	**1.399**	8.731	**8.943**	7.487
					Example 5 ($d = 0.5$)				
$(50, 50)$	**24.189**	23.935	21.005	11.916	13.405	**15.291**	36.105	**37.340**	36.296
$(50, 100)$	**23.858**	23.494	21.389	12.132	14.311	**16.480**	35.990	37.805	**37.869**
$(100, 100)$	**47.927**	47.217	43.802	25.852	30.221	**33.975**	73.780	77.439	**77.777**
$(100, 200)$	**47.730**	46.898	45.228	27.808	33.306	**35.603**	75.539	80.204	**80.831**
$(200, 200)$	**94.918**	93.398	89.976	55.774	66.264	**70.893**	150.692	159.662	**160.869**

From Table A.1, we can see that the CPCA provides feature $\widehat{k}_t$ with the largest variance, while the first step of our method (SWPCA) captures $\widehat{k}_t$ with the largest lag 1 auto-covariance. In addition, in *Example* 5, our method has slightly larger Mix($\widehat{k}_t$) than DPCA, which shows that under certain data structure the dimension reduction of our first step is enough to represent sufficient information.

From Table A.2 and A.3, we can see that our method always provides the error terms with the smallest time and cross-sectional variance and dependence.

Table A.4 and A.5, show the the 1 step and 5 steps ahead root mean square errors for *Example* 5 with $d = 0.5, 0.4, 0.3$, respectively. Overall, SWPCA performs better than the other two, as it has the smallest overall FRMSE for all the cases. Checking the Dependent FRMSE and Independent FRMSE separately, we can find that SWPCA performs even better for the dependent part. As d

Table A.2: Variance across Time and Ages of error terms

	Time Variance ($\widehat{\varepsilon}_{\cdot t}$)			Cross-sectional Variance ($\widehat{\varepsilon}_{p\cdot}$)		
(P,T)	CPCA	DPCA	SWPCA	CPCA	DPCA	SWPCA
			Example 4			
$(50,50)$	0.778	0.680	**0.337**	0.794	0.694	**0.343**
$(50,100)$	0.775	0.706	**0.373**	0.782	0.713	**0.377**
$(100,100)$	0.796	0.694	**0.351**	0.804	0.701	**0.355**
$(100,200)$	0.778	0.708	**0.381**	0.782	0.712	**0.383**
$(200,200)$	0.803	0.701	**0.357**	0.807	0.704	**0.359**
			Example 5 ($d = 0.5$)			
$(50,50)$	0.071	0.091	**0.037**	0.094	0.123	**0.037**
$(50,100)$	0.056	0.080	**0.038**	0.067	0.104	**0.038**
$(100,100)$	0.055	0.077	**0.039**	0.064	0.100	**0.039**
$(100,200)$	0.046	0.074	**0.039**	0.050	0.094	**0.039**
$(200,200)$	0.046	0.072	**0.039**	0.051	0.091	**0.039**

Table A.3: Covariance across Time and Ages of error terms

	Time dependence ($\widehat{\varepsilon}_{\cdot t}$)			Cross-sectional dependence ($\widehat{\varepsilon}_{p\cdot}$)		
(P,T)	CPCA	DPCA	SWPCA	CPCA	DPCA	SWPCA
			Example 4			
$(50,50)$	0.107	0.097	**0.059**	0.108	0.099	**0.060**
$(50,100)$	0.106	0.099	**0.064**	0.086	0.082	**0.058**
$(100,100)$	0.077	0.070	**0.043**	0.077	0.070	**0.043**
$(100,200)$	0.075	0.070	**0.046**	0.061	0.058	**0.041**
$(200,200)$	0.054	0.050	**0.031**	0.055	0.050	**0.031**
			Example 5 ($d = 0.5$)			
$(50,50)$	0.025	0.037	**0.004**	0.032	0.045	**0.004**
$(50,100)$	0.016	0.030	**0.004**	0.015	0.032	**0.003**
$(100,100)$	0.013	0.027	**0.003**	0.014	0.030	**0.003**
$(100,200)$	0.007	0.025	**0.003**	0.007	0.025	**0.002**
$(200,200)$	0.007	0.023	**0.002**	0.007	0.024	**0.002**

decreasing, the Dependent FRMSE of SWPCA increases the least, although all the Dependent FRMSEs increase. And for the independent part, SWPCA is better when $d = 0.5, 0.4$, but performs almost the same with others when $d = 0.3$. This result shows that SWPCA extracts features with more forecasting power from the dependent part and uses it to help improve the forecasting of the independent part. However, when the proportion of the dependent part is small, such as $d = 0.3$, forecasting the independent part cannot be blessed that much from the dependent part. Therefore, there will be very little difference among the three methods when comparing the performance for the independent part when $d = 0.3$. But SWPCA will always provide better forecasting for the dependent part, which leads to better overall forecasting for all cases.

Table A.4: 1 Step Ahead Forecasting RMSE

(P, T)	Dependent FRMSE(1)			Independent FRMSE(1)			Overall FRMSE(1)		
	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA
Example 5 ($d = 0.5$)									
$(50, 50)$	0.623	0.608	**0.568**	0.771	0.768	**0.749**	0.757	0.748	**0.716**
$(50, 100)$	0.573	0.567	**0.550**	0.746	0.747	**0.743**	0.715	0.711	**0.700**
$(100, 100)$	0.568	0.558	**0.536**	0.760	0.746	**0.734**	0.719	0.705	**0.688**
$(100, 200)$	0.572	0.568	**0.553**	0.782	0.772	**0.764**	0.732	0.725	**0.711**
$(200, 200)$	0.537	0.531	**0.521**	0.759	0.752	**0.746**	0.705	0.696	**0.686**
Example 5 ($d = 0.4$)									
$(50, 50)$	0.654	0.642	**0.584**	**0.738**	0.743	0.741	0.759	0.756	**0.731**
$(50, 100)$	0.603	0.595	**0.553**	0.727	0.731	**0.712**	0.732	0.731	**0.699**
$(100, 100)$	0.631	0.607	**0.562**	0.767	0.768	**0.754**	0.766	0.755	**0.727**
$(100, 200)$	0.575	0.580	**0.542**	0.778	0.774	**0.767**	0.752	0.752	**0.729**
$(200, 200)$	0.570	0.568	**0.537**	0.739	0.742	**0.733**	0.719	0.720	**0.701**
Example 5 ($d = 0.3$)									
$(50, 50)$	0.708	0.686	**0.598**	0.761	0.764	**0.760**	0.798	0.793	**0.760**
$(50, 100)$	0.687	0.653	**0.572**	**0.749**	0.752	**0.749**	0.782	0.772	**0.742**
$(100, 100)$	0.695	0.640	**0.556**	**0.773**	0.774	0.775	0.805	0.788	**0.757**
$(100, 200)$	0.682	0.606	**0.533**	**0.742**	0.747	0.743	0.772	0.749	**0.719**
$(200, 200)$	0.680	0.620	**0.549**	**0.719**	0.720	0.720	0.759	0.737	**0.711**

Table A.6 shows the the 1 step and 5 steps ahead root mean square errors of SWPCA compared to DPCA(ℓ), $\ell = 1, 5, 10$, for *Example* 6. The reason for

comparing DPCA(ℓ) separately is that it contains different information. The DPCA we compared with in *Example* 5 involves the same information (variance and lag 1 auto-covariance of $\boldsymbol{y}_t$) with SWPCA, while DPCA(ℓ) aggregates more dependent information (lag 1 to lag ℓ auto-covariances) but discards $var\,(\boldsymbol{y}_t)$. In addition, DPCA(1) is equivalent to only conduct the first step of SWPCA. In Table A.6, we can see that SWPCA and DPCA(1) perform better than DPCA(5) and DPCA(10) for most (P, T) cases. This shows that involving more lagged auto-covariances does not always provide more useful information for forecasting under certain situations. The performance of DPCA(1) is worse than SWPCA with $(P, T) = (50, 50), (50, 100), (100, 100)$ for 1 step ahead forecasting and $(P, T) = (100, 100)$ for 5 steps ahead forecasting, and similar for other cases. These results show that when variation is large, it is necessary to conduct the second step in the SWPCA in order to achieve more accurate forecasting.

Table A.5: 5 Steps Ahead Forecasting RMSE

(P,T)	Dependent FRMSE(5)			Independent FRMSE(5)			Overall FRMSE(5)		
	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA	CPCA	DPCA	SW-PCA
Example 5 ($d = 0.5$)									
$(50, 50)$	0.846	**0.839**	**0.839**	0.891	0.893	**0.879**	0.899	0.896	**0.888**
$(50, 100)$	0.833	0.834	**0.827**	0.876	0.870	**0.864**	0.883	0.881	**0.874**
$(100, 100)$	0.823	0.817	**0.811**	0.885	0.878	**0.867**	0.883	0.876	**0.866**
$(100, 200)$	0.802	0.798	**0.794**	0.872	0.866	**0.862**	0.866	0.861	**0.857**
$(200, 200)$	0.790	0.795	**0.788**	0.862	0.857	**0.854**	0.852	0.852	**0.847**
Example 5 ($d = 0.4$)									
$(50, 50)$	0.864	0.854	**0.838**	0.876	0.877	**0.867**	0.901	0.899	**0.886**
$(50, 100)$	0.826	0.815	**0.803**	0.875	0.877	**0.868**	0.885	0.882	**0.872**
$(100, 100)$	0.815	0.817	**0.807**	0.870	0.872	**0.860**	0.874	0.876	**0.865**
$(100, 200)$	0.787	0.781	**0.772**	0.866	0.865	**0.857**	0.859	0.856	**0.849**
$(200, 200)$	0.792	0.797	**0.786**	0.865	0.868	**0.852**	0.861	0.864	**0.851**
Example 5 ($d = 0.3$)									
$(50, 50)$	0.877	0.863	**0.835**	**0.885**	0.889	**0.885**	0.910	0.909	**0.896**
$(50, 100)$	0.860	0.841	**0.821**	**0.855**	0.856	**0.855**	0.883	0.878	**0.870**
$(100, 100)$	0.890	0.865	**0.833**	**0.847**	0.848	**0.847**	0.887	0.880	**0.868**
$(100, 200)$	0.831	0.804	**0.771**	0.854	0.856	**0.852**	0.871	0.864	**0.851**
$(200, 200)$	0.851	0.818	**0.801**	0.867	0.868	**0.866**	0.888	0.877	**0.869**

Table A.6: 1 step and 5 steps ahead RMSE, Example 6

(P,T)	Overall FRMSE(1)				Overall FRMSE(5)			
	DPCA(1)	DPCA(5)	DPCA(10)	SW-PCA	DPCA(1)	DPCA(5)	DPCA(10)	SW-PCA
$(50,50)$	1.371	1.379	1.376	**1.367**	1.518	**1.518**	**1.518**	1.527
$(50,100)$	1.312	1.332	1.332	**1.309**	**1.495**	**1.496**	**1.495**	1.497
$(100,100)$	1.389	1.403	1.402	**1.384**	1.490	1.490	1.490	**1.488**
$(100,200)$	**1.349**	1.371	1.370	**1.349**	**1.456**	1.462	1.462	1.456
$(200,200)$	**1.329**	1.355	1.355	**1.329**	**1.481**	1.489	1.489	**1.481**

A.2 Proof of Theorem 2.1

This section contains proof of Theorem 2.1 in Chapter 2, as well as some lemmas that are used in these proofs. Before introducing the proofs, we provide some notations. For a $k \times k$ matrix $\boldsymbol{F}$, $\lambda_i(\boldsymbol{F})$ indicates the i-th largest eigenvalue of the matrix $\boldsymbol{F}$. For a non-symmetric matrix $\boldsymbol{S}$, we use $\sigma_j(\boldsymbol{S})$ to denote the singular value of the matrix $\boldsymbol{S}$, which corresponds to the j-th largest eigenvalue of the matrix $\boldsymbol{S}\boldsymbol{S}^\top$. Let $||\boldsymbol{F}||$ be the square root of the maximum eigenvalue of $\boldsymbol{F}\boldsymbol{F}^\top$ and $||\boldsymbol{F}||_{\min}$ be the square root of the smallest nonzero eigenvalue of the matrix $\boldsymbol{F}\boldsymbol{F}^\top$. The notation $a \asymp b$ means that $a = O(b)$ and $b = O(a)$.

Useful Lemmas

We will introduce four lemmas that will be used in the proofs of Theorem 2.1. Lemma A.1, Lemma A.2 and Lemma A.3 are available results on eigenvalues of matrices under various decomposition. Lemma A.4 provides the orders of eigenvalues of the matrix $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$, and the proof follows up the statement of Lemma A.4.

Lemma A.1 (Weyl's Theorem)**.** *Let* $\{\lambda_i(\boldsymbol{S}) : i = 1,\dots,P\}$ *be eigenvalues of the matrix* $\boldsymbol{S}$ *in descending order and* $\{\lambda_i(\boldsymbol{J}) : i = 1,\dots,P\}$ *be eigenvalues of the matrix* $\boldsymbol{J}$ *in descending order. Then*

$$|\lambda_i(\boldsymbol{S}) - \lambda_i(\boldsymbol{J})| \leq ||\boldsymbol{S} - \boldsymbol{J}||. \tag{A.2.1}$$

Lemma A.2 (Lemma S.1 of Lam and Yao [2012]). *Let $\boldsymbol{F}$ be a $k \times k$ symmetric matrix such that*

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{G} & \boldsymbol{H} \\ \boldsymbol{H}^\top & \boldsymbol{D} \end{pmatrix} \tag{A.2.2}$$

with $\boldsymbol{G} : k_1 \times k_1$, $\boldsymbol{D} : k_2 \times k_2$ and $\lambda_{k_1}(\boldsymbol{G}) > \lambda_1(\boldsymbol{D})$. Note that $k_1 + k_2 = k$. Then for $1 \leq j \leq k_2$,

$$0 \leq \lambda_j(\boldsymbol{D}) - \lambda_{k_1+j}(\boldsymbol{F}) \leq \frac{\lambda_1\left(\boldsymbol{H}\boldsymbol{H}^\top\right)}{\lambda_{k_1}(\boldsymbol{G}) - \lambda_j(\boldsymbol{D})}. \tag{A.2.3}$$

Lemma A.3 (Lemma 3 of Lam et al. [2011]). *Suppose $\boldsymbol{F}$ and $\boldsymbol{F} + \boldsymbol{E}$ are $P \times P$ symmetric matrices and that $\boldsymbol{Q} = (\boldsymbol{Q}_1, \boldsymbol{Q}_2)$, where $\boldsymbol{Q}_1$ has size $P \times k$ and $\boldsymbol{Q}_2$ has size $P \times (P-k)$, is an orthogonal matrix such that span$(\boldsymbol{Q}_1)$ is an invariant subspace for the matrix $\boldsymbol{F}$, that is, $\boldsymbol{F} \times span(\boldsymbol{Q}_1) \subset span(\boldsymbol{F})$. Partition the matrices $\boldsymbol{Q}^\top \boldsymbol{F}\boldsymbol{Q}$ and $\boldsymbol{Q}^T \boldsymbol{E}\boldsymbol{Q}$ as follows.*

$$\boldsymbol{Q}^\top \boldsymbol{F}\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_2 \end{pmatrix}, \quad \boldsymbol{Q}^\top \boldsymbol{E}\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{E}_{11} & \boldsymbol{E}_{21}^\top \\ \boldsymbol{E}_{21} & \boldsymbol{E}_{22} \end{pmatrix}. \tag{A.2.4}$$

If $sep(\boldsymbol{D}_1, \boldsymbol{D}_2) := \min_{\lambda \in \Lambda(\boldsymbol{D}_1), \mu \in \Lambda(\boldsymbol{D}_2)} |\lambda - \mu| > 0$, where $\Lambda(\boldsymbol{D}_1)$ denotes the set of eigenvalues of the matrix $\boldsymbol{D}_1$ and $||\boldsymbol{E}|| \leq sep(\boldsymbol{D}_1, \boldsymbol{D}_2)/5$, then there exists a matrix $\boldsymbol{P} : (P-k) \times k$ with

$$||\boldsymbol{P}|| \leq \frac{4\,||\boldsymbol{E}_{21}||}{sep(\boldsymbol{D}_1, \boldsymbol{D}_2)} \tag{A.2.5}$$

such that the columns of the matrix $\widehat{\boldsymbol{Q}}_1 = (\boldsymbol{Q}_1 + \boldsymbol{Q}_2\boldsymbol{P})\left(\boldsymbol{I} + \boldsymbol{P}^\top \boldsymbol{P}\right)^{-1/2}$ define an orthogonal basis for a subspace that is invariant for the matrix $\boldsymbol{F} + \boldsymbol{E}$.

Lemma A.4. *Under Assumptions 2.1-2.8, we have*

$$\lambda_j\left(\boldsymbol{L}_1\right) \asymp P^{2-2\delta_1}, \quad j=1,\ldots,r_1. \tag{A.2.6}$$

$$\lambda_{r_1+j}\left(\boldsymbol{L}_1\right) \asymp P^{2-2\delta_2}, \quad j=1,\ldots,r_2; \tag{A.2.7}$$

$$\lambda_{r_1+r_2+i}\left(\boldsymbol{L}_1\right) = o_p\left(P^{1-\delta_1}\right), \quad i=1,\ldots,P-(r_1+r_2). \tag{A.2.8}$$

$$\lambda_i\left(\boldsymbol{L}_2\right) \asymp P^2, \quad i=1,\ldots,r_2. \tag{A.2.9}$$

proof of Lemma A.4. Recall the two-style factor model

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{k}_t^{(1)} + \boldsymbol{A}\boldsymbol{k}_t^{(2)} + \boldsymbol{\varepsilon}_t. \tag{A.2.10}$$

From the expression (A.2.10), the population covariance matrix of $\boldsymbol{y}_t$ has the following decomposition

$$\boldsymbol{\Sigma}_y(1) = \boldsymbol{B}\boldsymbol{M}_1 + \boldsymbol{A}\boldsymbol{M}_2 + \boldsymbol{\Sigma}_\varepsilon(1), \tag{A.2.11}$$

where

$$\boldsymbol{M}_1 = \boldsymbol{\Sigma}_k^{(1)}(1)\boldsymbol{B}^\top + \boldsymbol{\Sigma}_k^{(12)}(1)\boldsymbol{A}^\top, \quad \boldsymbol{M}_2 = \boldsymbol{\Sigma}_k^{(2)}(1)\boldsymbol{A}^\top + \boldsymbol{\Sigma}_k^{(21)}(1)\boldsymbol{B}^\top.$$

Based on Lemma A.1, we can evaluate the j-th eigenvalue of $\boldsymbol{L}_1$ below, $j = 1,\ldots,r_1$,

$$\begin{aligned}
\lambda_j\left(\boldsymbol{L}_1\right) &= \sigma_j^2\left(\boldsymbol{\Sigma}_y(1)\right) \geq \left[\sigma_j\left(\boldsymbol{B}\boldsymbol{M}_1\right) - \sigma_1\left(\boldsymbol{A}\boldsymbol{M}_2 + \boldsymbol{\Sigma}_\varepsilon(1)\right)\right]^2 \\
&\geq \left[\sigma_j\left(\boldsymbol{B}\boldsymbol{M}_1\right) - \sigma_1\left(\boldsymbol{A}\boldsymbol{M}_2\right) - \sigma_1\left(\boldsymbol{\Sigma}_\varepsilon(1)\right)\right]^2 \\
&= \left[\sigma_j\left(\boldsymbol{M}_1\right) - \sigma_1\left(\boldsymbol{M}_2\right) - \sigma_1\left(\boldsymbol{\Sigma}_\varepsilon(1)\right)\right]^2 \\
&\asymp \sigma^2\left(\boldsymbol{M}_1\right) \geq \left[\sigma_j\left(\boldsymbol{\Sigma}_k^{(1)}(1)\boldsymbol{B}^\top\right) - \sigma_1\left(\boldsymbol{\Sigma}_k^{(12)}(1)\boldsymbol{A}^\top\right)\right]^2 \\
&\asymp \sigma_j^2\left(\boldsymbol{\Sigma}_k^{(1)}(1)\boldsymbol{B}^\top\right) = \sigma_j^2\left(\boldsymbol{\Sigma}_k^{(1)}(1)\right) \geq \left\|\boldsymbol{\Sigma}_k^{(1)}(1)\right\|_{\min}^2 = P^{2-2\delta_1},
\end{aligned}$$

where the first and second inequalities use Lemma A.1; the second equality uses the matrices $\boldsymbol{B}$ and $\boldsymbol{A}$ being orthonormal assumed in Assumption 2.1; and the last inequality and equality both utilize Assumption 2.2.

Hence, the first r_1 largest eigenvalues of the matrix $\boldsymbol{L}_1$ have the order of $P^{2-2\delta_1}$.

Now we consider the order of the left $p-r_1$ eigenvalues of the matrix $\boldsymbol{L}_1$. In terms of Weyl's inequality in Lemma A.1, we use the eigenvalues of the matrix $\widetilde{\boldsymbol{L}}_1 = \widetilde{\boldsymbol{\Sigma}}_y(1)\widetilde{\boldsymbol{\Sigma}}_y(1)$ to approximate the eigenvalues of $\boldsymbol{L}_1$, where $\widetilde{\boldsymbol{\Sigma}}_y(1) = \boldsymbol{B}\boldsymbol{M}_1 + \boldsymbol{A}\boldsymbol{M}_2$. In fact,

$$\begin{aligned}
&\left|\lambda_{r_1+j}\left(\boldsymbol{L}_1\right) - \lambda_{r_1+j}\left(\widetilde{\boldsymbol{L}}_1\right)\right| \leq \left|\left|\boldsymbol{L}_1 - \widetilde{\boldsymbol{L}}_1\right|\right| \\
&\leq ||\boldsymbol{B}\boldsymbol{M}_1 + \boldsymbol{A}\boldsymbol{M}_2 + \boldsymbol{\Sigma}_\varepsilon(1)|| \cdot ||\boldsymbol{\Sigma}_\varepsilon(1)|| + ||\boldsymbol{B}\boldsymbol{M}_1 + \boldsymbol{A}\boldsymbol{M}_2|| \cdot ||\boldsymbol{\Sigma}_\varepsilon(1)|| \\
&= o\left(P^{1-\delta_1}\right), \qquad \text{(A.2.12)}
\end{aligned}$$

where the last equality uses Assumption 2.2.

Now we evaluate the order of $\lambda_{r_1+j}\left(\widetilde{\boldsymbol{L}}_1\right)$. Note that the rank of $\widetilde{\boldsymbol{L}}_1$ is no larger than $r_1 + r_2$. So, when $j > r_1 + r_2$, $\lambda_{r_1+j}\left(\widetilde{\boldsymbol{L}}_1\right) = 0$. Hence, next we investigate the case of $j = 1, \ldots, r_2$.

Decompose $\widetilde{\boldsymbol{L}}_1$ in the following way.

$$\begin{aligned}
\widetilde{\boldsymbol{L}}_1 &= \begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}_1 \\ \boldsymbol{M}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{M}_1^\top & \boldsymbol{M}_2^\top \end{pmatrix} \begin{pmatrix} \boldsymbol{B}^\top \\ \boldsymbol{A}^\top \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}_1\boldsymbol{M}_1^\top & \boldsymbol{M}_1\boldsymbol{M}_2^\top \\ \boldsymbol{M}_2\boldsymbol{M}_1^\top & \boldsymbol{M}_2\boldsymbol{M}_2^\top \end{pmatrix} \begin{pmatrix} \boldsymbol{B}^\top \\ \boldsymbol{A}^\top \end{pmatrix}. \qquad \text{(A.2.13)}
\end{aligned}$$

Because

$$\begin{pmatrix} \boldsymbol{B}^\top \\ \boldsymbol{C}^\top \end{pmatrix} \begin{pmatrix} \boldsymbol{B} & \boldsymbol{C} \end{pmatrix} = \boldsymbol{I}, \qquad \text{(A.2.14)}$$

we have $\lambda_j(\boldsymbol{L}_1) = \lambda_j(\boldsymbol{M})$, where

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{M}_1\boldsymbol{M}_1^\top & \boldsymbol{M}_1\boldsymbol{M}_2^\top \\ \boldsymbol{M}_2\boldsymbol{M}_1^\top & \boldsymbol{M}_2\boldsymbol{M}_2^\top \end{pmatrix}. \tag{A.2.15}$$

It follows from Lemma A.3 and Assumption 2.2 that

$$\lambda_{r_1+j}(\boldsymbol{M}) \leq \lambda_j\left(\boldsymbol{M}_2\boldsymbol{M}_2^\top\right) \asymp P^{2-2\delta_2}, \tag{A.2.16}$$

and

$$\lambda_{r_1+j}(\boldsymbol{M}) \geq \sigma_j^2(\boldsymbol{M}_2) - \frac{\sigma_1^2\left(\boldsymbol{M}_1\boldsymbol{M}_2^\top\right)}{\sigma_{r_1}^2(\boldsymbol{M}_1) - \sigma_j^2(\boldsymbol{M}_2)} \asymp P^{2-2\delta_2}, \tag{A.2.17}$$

where the last $\asymp$ above uses the fact that $\sigma_{r_1}^2(\boldsymbol{M}_1) \asymp P^{2-2\delta_1}$, $\sigma_j^2(\boldsymbol{M}_2) \asymp P^{2-2\delta_2}$ and $\sigma_1^2\left(\boldsymbol{M}_1\boldsymbol{M}_2^\top\right) = O\left(P^{4-2(\delta_1+\delta_2)}\right)$.

Combining (A.2.16) and (A.2.17), we can get

$$\lambda_{r_1+j}(\boldsymbol{M}) \asymp P^{2-2\delta_2}, \quad j = 1, \ldots, r_2. \tag{A.2.18}$$

Then it follows from (A.2.12), (A.2.18) and Assumption 2.2 that

$$\lambda_{r_1+j}(\boldsymbol{L}_1) \asymp P^{2-2\delta_2}, \quad j = 1, \ldots, r_2; \tag{A.2.19}$$

$$\lambda_{r_1+r_2+i}(\boldsymbol{L}_1) = o_p\left(P^{1-\delta_1}\right), \quad i = 1, \ldots, P-(r_1+r_2). \tag{A.2.20}$$

Finally, the order of $\lambda_i(\boldsymbol{L}_2)$ can be derived from Proposition 2.1 of Fan et al. [2013] directly. □

Proof of Theorem 2.1

Proof of Theorem 2.1. Let $\boldsymbol{E}_L^{(1)} = \widehat{\boldsymbol{L}}_1 - \boldsymbol{L}_1$ with $\widehat{\boldsymbol{L}}_1 = \widehat{\boldsymbol{\Sigma}}_y(1)\widehat{\boldsymbol{\Sigma}}_y(1)^\top$ and $\boldsymbol{L}_1 = \boldsymbol{\Sigma}_y(1)\boldsymbol{\Sigma}_y(1)^\top$. First we evaluate the order of $\left\|\boldsymbol{E}_L^{(1)}\right\|$. In terms of simple calcula-

tions, we have

$$\left\|\boldsymbol{E}_L^{(1)}\right\| \leq \left\|\widehat{\boldsymbol{\Sigma}}_y(1) - \boldsymbol{\Sigma}_y(1)\right\|^2 + 2\left\|\widehat{\boldsymbol{\Sigma}}_y(1)\right\| \cdot \left\|\widehat{\boldsymbol{\Sigma}}_y(1) - \boldsymbol{\Sigma}_y(1)\right\|. \tag{A.2.21}$$

In terms of (A.2.6) in Lemma A.4, we have $||\boldsymbol{\Sigma}_y(1)|| \asymp P^{1-\delta_1}$. From (A.2.11), we can get

$$\begin{aligned}\left\|\widehat{\boldsymbol{\Sigma}}_y(1) - \boldsymbol{\Sigma}_y(1)\right\| \leq \left\|\widehat{\boldsymbol{M}}_1 - \boldsymbol{M}_1\right\| + \left\|\widehat{\boldsymbol{M}}_2 - \boldsymbol{M}_2\right\| \\ + \left\|\widehat{\boldsymbol{\Sigma}}_\varepsilon(1) - \boldsymbol{\Sigma}_\varepsilon(1)\right\|,\end{aligned} \tag{A.2.22}$$

where

$$\widehat{\boldsymbol{M}}_1 = \widehat{\boldsymbol{\Sigma}}_k^{(1)}(1)\boldsymbol{B}^\top + \widehat{\boldsymbol{\Sigma}}_k^{(12)}(1)\boldsymbol{A}^\top, \quad \widehat{\boldsymbol{M}}_2 = \widehat{\boldsymbol{\Sigma}}_k^{(2)}(1)\boldsymbol{A}^\top + \widehat{\boldsymbol{\Sigma}}_k^{(21)}(1)\boldsymbol{B}^\top,$$

with $\widehat{\boldsymbol{\Sigma}}_k^{(1)}(1)$, $\widehat{\boldsymbol{\Sigma}}_k^{(12)}(1)$, $\widehat{\boldsymbol{\Sigma}}_k^{(2)}(1)$ and $\widehat{\boldsymbol{\Sigma}}_k^{(21)}(1)$ are the sample covariances corresponding to the population covariances $\boldsymbol{\Sigma}_k^{(1)}(1)$, $\boldsymbol{\Sigma}_k^{(12)}(1)$, $\boldsymbol{\Sigma}_k^{(2)}(1)$ and $\boldsymbol{\Sigma}_k^{(21)}(1)$, respectively.

Hence, we evaluate (A.2.22) further

$$\begin{aligned}\left\|\widehat{\boldsymbol{\Sigma}}_y(1) - \boldsymbol{\Sigma}_y(1)\right\| &\leq \left\|\widehat{\boldsymbol{\Sigma}}_k^{(1)}(1) - \boldsymbol{\Sigma}_k^{(1)}(1)\right\| + \left\|\widehat{\boldsymbol{\Sigma}}_k^{(12)}(1) - \boldsymbol{\Sigma}_k^{(12)}(1)\right\| \\ &\quad + \left\|\widehat{\boldsymbol{\Sigma}}_k^{(2)}(1) - \boldsymbol{\Sigma}_k^{(2)}(1)\right\| + \left\|\widehat{\boldsymbol{\Sigma}}_k^{(21)}(1) - \boldsymbol{\Sigma}_k^{(21)}(1)\right\| \\ &\quad + \left\|\widehat{\boldsymbol{\Sigma}}_\varepsilon(1) - \boldsymbol{\Sigma}_\varepsilon(1)\right\| \\ &= O_p\left(\frac{P^{1-\delta_1}}{T^{1/2}}\right) + O_p\left(\frac{P^{1-\delta_2}}{T^{1/2}}\right) + O_p\left(\frac{P}{T}\right) \\ &= O_p\left(\max\left(\frac{P}{T}, \frac{P^{1-\delta_1}}{T^{1/2}}\right)\right),\end{aligned} \tag{A.2.23}$$

where the last second equality uses (A8) of Lam et al. [2011] which demonstrates $\left\|\widehat{\boldsymbol{\Sigma}}_\varepsilon(1) - \boldsymbol{\Sigma}_\varepsilon(1)\right\| = O_p\left(\frac{P}{T}\right)$.

Then it follows from (A.2.21) and (A.2.23) that

$$\left\|\boldsymbol{E}_L^{(1)}\right\| = O_p\left(\max\left(\frac{P^2}{T^2}, \frac{P^{2-2\delta_1}}{T}, \frac{P^{2-\delta_1}}{T}, \frac{P^{2-2\delta_1}}{T^{1/2}}\right)\right) \tag{A.2.24}$$

$$= O_p\left(\frac{P^{2-2\delta_1}}{T^{1/2}}\right), \tag{A.2.25}$$

where the last equality uses the assumption that $P^{\delta_1} = o\left(T^{1/2}\right)$.

Now we use Lemma A.3 to get the order of estimated factor loadings. In Lemma A.3, let $\boldsymbol{F}$ and $\boldsymbol{E}$ be $\boldsymbol{L}_1$ and $\widehat{\boldsymbol{L}}_1 - \boldsymbol{L}_1$, respectively. Let k in Lemma 3 equal to r_1. Then we have, from (A.2.6),

$$sep(\boldsymbol{D}_1, \boldsymbol{D}_2) \asymp P^{2-2\delta_1}, \tag{A.2.26}$$

where the defition of $sep(\cdot,\cdot)$ is provided in Lemma A.3. Then $\boldsymbol{E}_L^{(1)}$ and $sep(\boldsymbol{D}_1, \boldsymbol{D}_2)$ satisfies

$$\left\|\boldsymbol{E}_L^{(1)}\right\| = o_p\left(sep(\boldsymbol{D}_1, \boldsymbol{D}_2)\right) \leq \frac{sep(\boldsymbol{D}_1, \boldsymbol{D}_2)}{5}. \tag{A.2.27}$$

Hence Lemma A.3 tells us that, there exists a matrix $\boldsymbol{P} : (P - r_1) \times r_1$ such that

$$||\boldsymbol{P}|| \leq \frac{4}{sep\left(\boldsymbol{D}_1, \boldsymbol{D}_2\right)} \cdot \left\|\left(\boldsymbol{E}_L^{(1)}\right)_{21}\right\| \leq \frac{4\left\|\boldsymbol{E}_L^{(1)}\right\|}{sep(\boldsymbol{D}_1, \boldsymbol{D}_2)} \tag{A.2.28}$$

and then $\widehat{\boldsymbol{B}} = \left(\boldsymbol{B} + \boldsymbol{B}^c\boldsymbol{P}\right)\left(\boldsymbol{I} + \boldsymbol{P}^\top\boldsymbol{P}\right)^{-1/2}$ is an estimator of $\boldsymbol{B}$ with $\boldsymbol{B}^c$ being $\boldsymbol{Q}_2$ in Lemma A.3. In view of this, the rate of convergence for $\widehat{\boldsymbol{B}}$ can be calculated

as

$$\begin{aligned}\left\|\widehat{\boldsymbol{B}}-\boldsymbol{B}\right\| &= \left\|\left(\boldsymbol{B}+\boldsymbol{B}^{c}\boldsymbol{P}\right)\left(\boldsymbol{I}+\boldsymbol{P}^{T}\boldsymbol{P}\right)^{-1/2}-\boldsymbol{B}\right\| \\ &= \left\|\left[\left(\boldsymbol{B}+\boldsymbol{B}\boldsymbol{P}\right)-\boldsymbol{B}\left(\boldsymbol{I}+\boldsymbol{P}^{\top}\boldsymbol{P}\right)^{1/2}\right]\left(\boldsymbol{I}+\boldsymbol{P}^{\top}\boldsymbol{P}\right)^{-1/2}\right\| \\ &= \left\|\left(\boldsymbol{B}\left[\boldsymbol{I}-\left(\boldsymbol{I}+\boldsymbol{P}^{\top}\boldsymbol{P}\right)^{1/2}\right]+\boldsymbol{B}^{c}\boldsymbol{P}\right)\left(\boldsymbol{I}+\boldsymbol{P}^{\top}\boldsymbol{P}\right)^{-1/2}\right\| \\ &\leq \left\|\boldsymbol{I}-\left(\boldsymbol{I}+\boldsymbol{P}^{\top}\boldsymbol{P}\right)^{1/2}\right\|+\left\|\boldsymbol{P}\right\| \leq 2\left\|\boldsymbol{P}\right\|, \end{aligned} \tag{A.2.29}$$

where the last second equality uses the fact that $\boldsymbol{B}$ and $\boldsymbol{B}^{c}$ are orthonormal; and the last equality uses the fact that

$$\begin{aligned}\left\|\boldsymbol{I}-\left(\boldsymbol{I}+\boldsymbol{P}^{\top}\boldsymbol{P}\right)^{1/2}\right\| &= 1-\left(1+\lambda_{\min}\left(\boldsymbol{P}^{\top}\boldsymbol{P}\right)\right)^{1/2} &\text{(A.2.30)}\\ &\leq \lambda_{\max}^{1/2}\left(\boldsymbol{P}^{\top}\boldsymbol{P}\right). &\text{(A.2.31)}\end{aligned}$$

Therefore, by (A.2.29) and (A.2.28), we obtain

$$\left\|\widehat{\boldsymbol{B}}-\boldsymbol{B}\right\| = O_{P}\left(\frac{\left\|\boldsymbol{E}_{L}^{(1)}\right\|}{sep(\boldsymbol{D}_{1},\boldsymbol{D}_{2})}\right) = O_{p}\left(\frac{1}{T^{1/2}}\right). \tag{A.2.32}$$

For the second factor model part, the estimation is to conduct principal component analysis on the residual of the first step, i.e. estimating the factor model

$$\widehat{\boldsymbol{u}}_{t} = \boldsymbol{A}\boldsymbol{k}_{t}^{(2)}+\boldsymbol{\eta}_{t}, \quad t=1,2,\ldots,T, \tag{A.2.33}$$

where $\widehat{\boldsymbol{u}}_{t}=\boldsymbol{y}_{t}-\widehat{\boldsymbol{B}}\widehat{\boldsymbol{k}}_{t}^{(1)}$, $\boldsymbol{\eta}_{t}$ is the new error component in the estimation at the second step.

In order to derive the rate of convergence for $\widehat{\boldsymbol{A}}$, we also utilize Lemma A.3. Now let $\boldsymbol{F}$ and $\boldsymbol{E}$ in Lemma A.3 are $\boldsymbol{L}_{2}$ and $\boldsymbol{E}_{L}^{(2)}:=\widehat{\boldsymbol{L}}_{2}-\boldsymbol{L}_{2}$, respectively. Let k in Lemma A.3 equal to r_{2}.

First, we evaluate $\left\|\boldsymbol{E}_L^{(2)}\right\|$. Based on (A.2.33), we have

$$\left\|\boldsymbol{E}_L^{(2)}\right\| \leq \left\|\widehat{\boldsymbol{\Sigma}}_{\widehat{u}}(0) - \boldsymbol{\Sigma}_u(0)\right\|^2 + 2\,||\boldsymbol{\Sigma}_u(0)|| \cdot \left\|\widehat{\boldsymbol{\Sigma}}_{\widehat{u}}(0) - \boldsymbol{\Sigma}_u(0)\right\|, \tag{A.2.34}$$

where $\boldsymbol{\Sigma}_u(0)$ is the population covariance matrix of $\boldsymbol{u}_t$ and $\widehat{\boldsymbol{\Sigma}}_{\widehat{u}}(0)$ is the sample covariance matrix of $\widehat{\boldsymbol{u}}_t$. Based on Assumption 2.3 and Proposition 2.1 of Fan et al. [2013], we know that $||\boldsymbol{\Sigma}_u(0)|| \asymp p$. For the term $\left\|\widehat{\boldsymbol{\Sigma}}_{\widehat{u}}(0) - \boldsymbol{\Sigma}_u(0)\right\|$, we evaluate its order as follows.

$$\begin{aligned}\left\|\widehat{\boldsymbol{\Sigma}}_{\widehat{u}}(0) - \boldsymbol{\Sigma}_u(0)\right\| &\leq \left\|\widehat{\boldsymbol{\Sigma}}_{\widehat{u}}(0) - \widehat{\boldsymbol{\Sigma}}_u(0)\right\| + \left\|\widehat{\boldsymbol{\Sigma}}_u(0) - \boldsymbol{\Sigma}_u(0)\right\| \\ &\leq \frac{1}{T}\left\|\widehat{\boldsymbol{B}}\widehat{\boldsymbol{K}}^{(1)} - \boldsymbol{B}\boldsymbol{K}^{(1)}\right\|^2 + \frac{2}{T}\left\|\boldsymbol{B}\boldsymbol{K}^{(1)}\right\| \cdot \left\|\widehat{\boldsymbol{B}}\widehat{\boldsymbol{K}}^{(1)} - \boldsymbol{B}\boldsymbol{K}^{(1)}\right\| \\ &\quad + \left\|\widehat{\boldsymbol{\Sigma}}_u(0) - \boldsymbol{\Sigma}_u(0)\right\|, \end{aligned} \tag{A.2.35}$$

where $\widehat{\boldsymbol{K}}^{(1)} = \left(\widehat{\boldsymbol{k}}_1^{(1)}, \widehat{\boldsymbol{k}}_2^{(1)}, \ldots, \widehat{\boldsymbol{k}}_T^{(1)}\right)$ and $\boldsymbol{K}^{(1)} = \left(\boldsymbol{k}_1^{(1)}, \boldsymbol{k}_2^{(1)}, \ldots, \boldsymbol{k}_T^{(1)}\right)$.

From Assumption 2.2 and (A.2.32), it can be derived that

$$\begin{aligned}\frac{1}{\sqrt{T}}\left\|\widehat{\boldsymbol{B}}\widehat{\boldsymbol{K}}^{(1)} - \boldsymbol{B}\boldsymbol{K}^{(1)}\right\| &= O_p\left(\frac{P^{1/2-\delta_1/2}}{\sqrt{T}}\right), \\ \frac{1}{\sqrt{T}}\left\|\boldsymbol{B}\boldsymbol{K}^{(1)}\right\| &= O_p\left(P^{1/2-\delta_1/2}\right). \end{aligned} \tag{A.2.36}$$

Similar to (A.2.23), we can also get

$$\begin{aligned}\left\|\widehat{\boldsymbol{\Sigma}}_u(0) - \boldsymbol{\Sigma}_u(0)\right\| &\leq \left\|\widehat{\boldsymbol{\Sigma}}_k^{(2)}(0) - \boldsymbol{\Sigma}_k^{(2)}(0)\right\| + \left\|\widehat{\boldsymbol{\Sigma}}_\varepsilon(0) - \boldsymbol{\Sigma}_\varepsilon(0)\right\| \\ &= O_p\left(\frac{P}{T^{1/2}}\right) + O_p\left(\frac{P}{T}\right). \end{aligned} \tag{A.2.37}$$

In view of (A.2.36), (A.2.37) and (A.2.35), we can get

$$\begin{aligned}\left\|\widehat{\boldsymbol{\Sigma}}_{\widehat{u}}(0)-\boldsymbol{\Sigma}_u(0)\right\| &= O_p\left(\frac{p^{1-\delta_1}}{\sqrt{T}}\right)+O_p\left(\frac{P}{\sqrt{T}}\right)+O_p\left(\frac{P}{T}\right)\\ &= O_p\left(\frac{P}{\sqrt{T}}\right).\end{aligned} \tag{A.2.38}$$

The order of $\left\|\boldsymbol{E}_L^{(2)}\right\|$ is obtained from (A.2.34) and (A.2.38), i.e.

$$\left\|\boldsymbol{E}_L^{(2)}\right\| = O_p\left(\frac{P^2}{\sqrt{T}}\right). \tag{A.2.39}$$

Moreover, it follows from Proposition 2.1 of Fan et al. [2013] that

$$sep(\boldsymbol{D}_1,\boldsymbol{D}_2) \asymp P^2. \tag{A.2.40}$$

Here $\boldsymbol{D}_1$ in Lemma A.3 is the diagonal matrix corresponding to the orthogonal matrix $\boldsymbol{A}$. Then we can get from Lemma A.3 that

$$\left\|\widehat{\boldsymbol{A}}-\boldsymbol{A}\right\| = O_p\left(\frac{\left\|\boldsymbol{E}_L^{(2)}\right\|}{sep\left(\boldsymbol{D}_2,\boldsymbol{D}_1\right)}\right) = O_p\left(\frac{1}{\sqrt{T}}\right). \tag{A.2.41}$$

□

Appendix of Chapter 3

B.1 Forecasting results for the gender-age-specific mortality rates of the US

Our analysis of the mortality forecasting in the main text focuses on age-specific US mortality data of the total population. This appendix provides additional gender-specific results of the estimation and out-of-sample forecasting for both males and females.

Male

The results for the male subpopulation are shown in Figure B.1 and Figure B.2.

In terms of the goodness of fit, the overall MSE for the classical factor model is 0.008479112, while the MSE for the time-varying model is 0.002679128. From the perspective of the out-of-sample forecasting precision, the overall MSPE for the classical factor model is 0.0412585, while if we choose to use naive method, MSPE for the time-varying model is only 0.02247346, which is approximately 45% less than the previous one. Based on local linear regression, the MSPE of time-varying model is 0.06222522, which is larger than that for the classical model. Thus, for the male subpopulation, compared with the classical model, the prediction accuracy improves a lot by using the time-varying factor model based on naive method. For the short-term forecasting, the naive method and local regression method have similar performances. However, for the long-term

forecasting, the prediction accuracy of local regression method deteriorates as time goes by. For the age-specific forecasts, the time-varying model based on naive method almost always obtains the best predictions. And roughly speaking, no matter which method we choose to use, the time-varying factor model is better at predicting the mortality rates of the older adulthood ($50 \sim 90$).

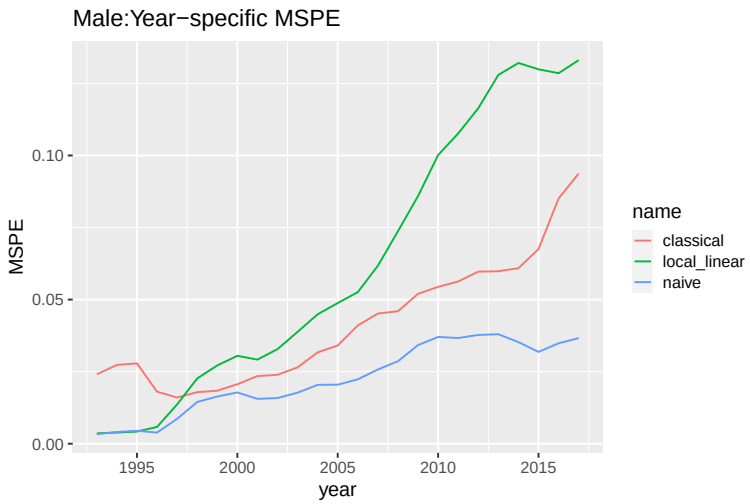


Figure B.1: Year-specific MSPE for both the time-varying model and the classical model; for time-varying model, both naive method and local regression method are used; **Male** subpopulation.

Female

The results for the female subpopulation are shown in Figure B.3 and Figure B.4.

In terms of the goodness of fit, the overall MSE for the classical factor model is 0.007337257, while the MSE for the time-varying model is 0.002061072. From the perspective of the out-of-sample forecasting precision, the overall MSPE for the classical factor model is 0.03709366, while if we choose to use naive method, MSPE for the time-varying model is 0.02962542, which is 20% less than the previous one. Based on local linear regression, the MSPE of time-varying model is 0.03887248, which is similar to (or slightly larger than) that for the classical model. Thus, for the female subpopulation, compared with the classical model,

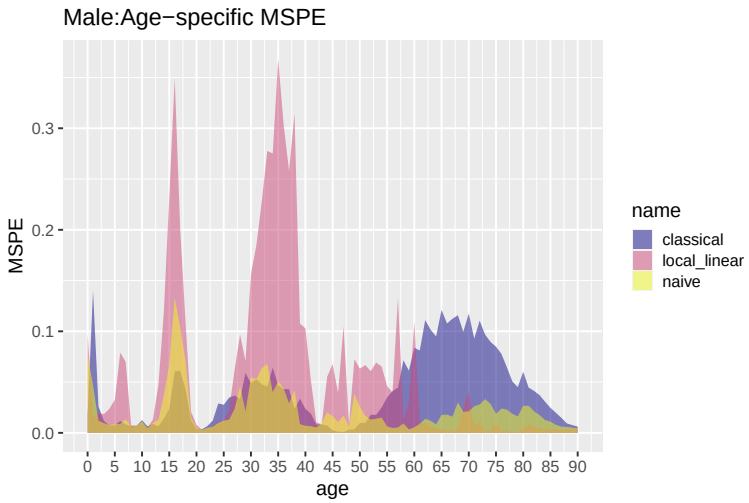


Figure B.2: Age-specific MSPE for both the time-varying model and the classical model; for time-varying model, both naive method and local regression method are used; **Male** subpopulation.

the prediction accuracy improves a lot by using the time-varying factor model based on naive method. For the short-term forecasting, the naive method and local regression method have similar performances. However, for the long-term forecasting, the prediction accuracy of local regression method deteriorates as time goes by. For the age-specific forecasts, the time-varying model based on naive method almost always obtains the best predictions, except for the age group $40 \sim 55$. And roughly speaking, no matter which method we use, the time-varying factor model is better at predicting the mortality rates of the young children ($0 \sim 10$), young adulthood ($20 \sim 40$) and the older adulthood ($55 \sim 80$).

In summary, the factor models (both the time-varying and the classical factor models) can capture the characteristics of the male mortality rates of the US better than the male mortality rates. Besides, compared with other methods, by using naive method for extrapolating factor loading, the prediction accuracy can be improved significantly based on the time-varying factor model, both for the male and female subpopulations.

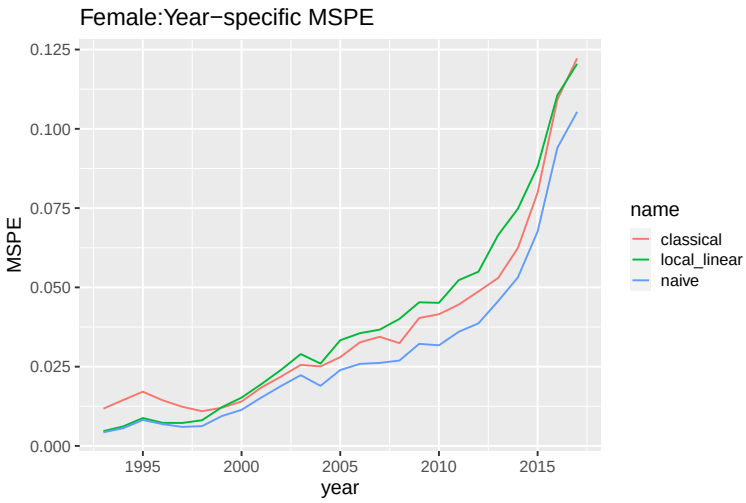


Figure B.3: Year-specific MSPE for both the time-varying model and the classical model; for time-varying model, both naive method and local regression method are used; **Female** subpopulation.

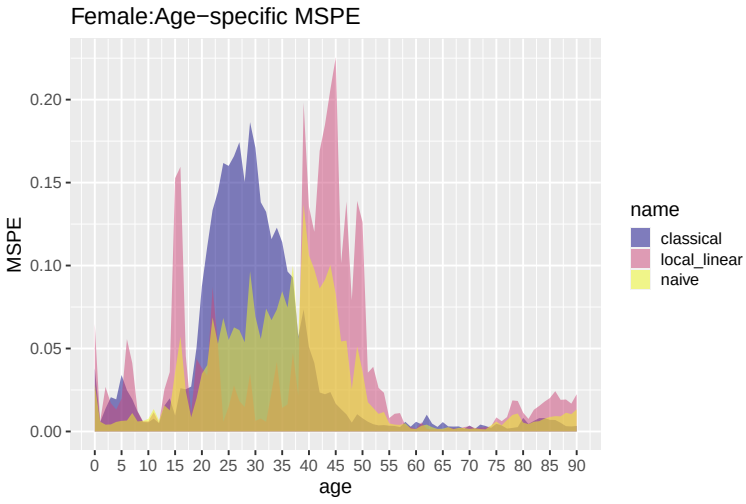


Figure B.4: Age-specific MSPE for both the time-varying model and the classical model; for time-varying model, both naive method and local regression method are used; **Female** subpopulation.

B.2 The estimated optimal "boundary" for multiple countries and fitting models

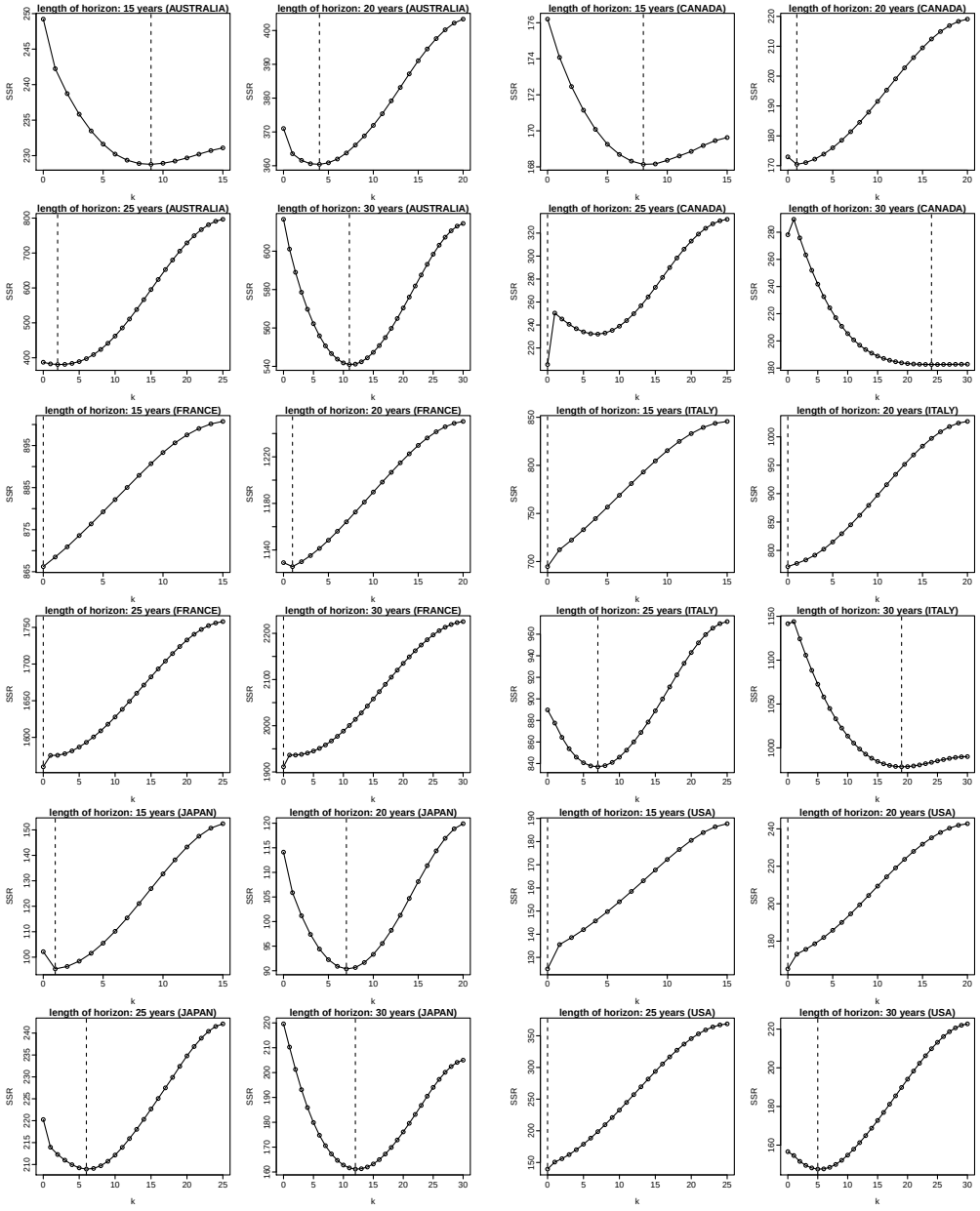


Figure B.5: Plots of the total sum of squared residuals (SSR) versus the length (k) of the short-term forecast horizon (based on the hybrid forecasting method of time-varying factor model); length of forecast horizon: 15, 20, 25 and 30

Bibliography

Ahn, S. C. and Horenstein, A. R., 2013. Eigenvalue ratio test for the number of factors. *Econometrica*, 81 (2013), 1203–1227. (cited on pages 30 and 34)

Anderson, T. W., 1963. The use of factor analysis in the statistical analysis of multiple time series. *Psychometrika*, 28, 1 (Mar 1963), 1–25. (cited on pages 6 and 61)

Anderson, T. W., 2003. *An Introduction to Multivariate Statistical Analysis*. Wiley, 3 edn. (cited on pages 18 and 104)

Ando, T. and Bai, J., 2017. Clustering huge number of financial time series: A panel data approach with high-dimensional predictors and factor structures. *Journal of the American Statistical Association*, 112, 519 (2017), 1182–1198. (cited on pages 1, 5, and 7)

Avella-Medina, M.; Battey, H. S.; Fan, J.; and Li, Q., 2018. Robust estimation of high-dimensional covariance and precision matrices. *Biometrika*, 105, 2 (2018), 271–284. (cited on page 11)

Bai, J., 2002. Determine the number of factors in approximate factor models. *Econometrica*, 70, 1 (2002), 191–221. (cited on pages 21, 22, and 23)

Bai, J., 2009. Panel data models with interactive fixed effects. *Econometrica*, 77, 4 (2009), 1229–1279. (cited on page 61)

Bai, J., 2010. Common breaks in means and variances for panel data. *Journal of Econometrics*, 157, 1 (2010), 78–92. (cited on pages 64 and 76)

Bai, J. and Ng, S., 2002. Determining the number of factors in approximate factor models. *Econometrica*, 70, 1 (2002), 191–221. (cited on pages 6 and 61)

Bell, W. R., 1997. Comparing and assessing time series methods for forecasting age-specific fertility and mortality rates. *Journal of Official Statistics*, 13, 3 (1997), 279. (cited on page 63)

Booth, H.; Hyndman, R. J.; Tickle, L.; and De Jong, P., 2006. Lee-Carter mortality forecasting: a multi-country comparison of variants and extensions. *Demographic Research*, 15 (2006), 289–310. (cited on page 62)

Booth, H.; Maindonald, J.; and Smith, L., 2002. Applying Lee-Carter under conditions of variable mortality decline. *Population Studies*, 56, 3 (2002), 325–336. (cited on pages 10, 62, and 75)

Booth, H. and Tickle, L., 2008. Mortality modelling and forecasting: A review of methods. *Annals of Actuarial Science*, 3, 1-2 (2008), 3–43. (cited on pages 9, 16, 17, and 62)

Booth, H.; Tickle, L.; et al., 2004. Beyond three score years and ten: prospects for longevity in Australia. *People and Place*, 12, 1 (2004), 15. (cited on page 62)

Bouwmans, T.; Sobral, A.; Javed, S.; Jung, S. K.; and Zahzah, E.-H., 2017. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 23 (2017), 1–71. (cited on page 13)

Breitung, J. and Eickmeier, S., 2011. Testing for structural breaks in dynamic factor models. *Journal of Econometrics*, 163, 1 (2011), 71 – 84. (cited on page 63)

Brillinger, D. R., 1975. *Time Series: Data Analysis and Theory*. Holt, Rinehart, and Winston. (cited on pages 7, 8, 18, 31, and 32)

Brockwell, P. J.; Davis, R. A.; and Fienberg, S. E., 1991. *Time Series: Theory and Methods: Theory and Methods*. Springer Science & Business Media. (cited on page 72)

Cai, T.; Han, X.; and Pan, G., 2017. Limiting laws for divergent spiked eigenvalues and largest non-spiked eigenvalue of sample covariance matrices. *arXiv preprint arXiv:1711.00217*, (2017). (cited on pages 105 and 118)

Cairns, A. J.; Blake, D.; and Dowd, K., 2006. A two-factor model for stochastic mortality with parameter uncertainty: theory and calibration. *Journal of Risk and Insurance*, 73, 4 (2006), 687–718. (cited on pages 5 and 10)

Cairns, A. J. G.; Blake, D.; Dowd, K.; Coughlan, G. D.; Epstein, D.; Ong, A.; and Balevich, I., 2009. A quantitative comparison of stochastic mortality models using data from England and Wales and the United States. *North American Actuarial Journal*, 13, 1 (2009), 1–35. (cited on page 63)

Campbell, N. A., 1980. Robust procedures in multivariate analysis i: Robust covariance estimation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 29, 3 (1980), 231–237. (cited on page 11)

Candès, E. J.; Li, X.; Ma, Y.; and Wright, J., 2011. Robust principal component analysis? *Journal of the ACM (JACM)*, 58, 3 (2011), 11. (cited on pages 13 and 104)

Chang, J.; Guo, B.; and Yao, Q., 2018. Principal component analysis for second-order stationary vector time series. *The Annals of Statistics*, 46, 5 (2018), 2094–2124. (cited on pages 18, 32, and 61)

Chen, L.; Dolado, J. J.; and Gonzalo, J., 2014. Detecting big structural breaks in large factor models. *Journal of Econometrics*, 180, 1 (2014), 30 – 48. (cited on page 63)

Chen, M.; Gao, C.; Ren, Z.; et al., 2018. Robust covariance and scatter matrix estimation under huber's contamination model. *The Annals of Statistics*, 46, 5 (2018), 1932–1960. (cited on page 11)

CMI, 2009. Continuous Mortality Investigation: A prototype mortality projections model: part two – detailed analysis. Working Paper 39, The Institute of Actuaries and the Faculty of Actuaries. (cited on page 64)

Croux, C. and Haesbroeck, G., 2000. Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87, 3 (2000), 603–618. (cited on pages 11 and 104)

Croux, C. and Ruiz-Gazen, A., 1996. A fast algorithm for robust principal components based on projection pursuit. In *Compstat*, 211–216. Springer. (cited on page 12)

Croux, C. and Ruiz-Gazen, A., 2005. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95, 1 (2005), 206–226. (cited on page 12)

Cui, H.; He, X.; and Ng, K. W., 2003. Asymptotic distributions of principal components based on robust dispersions. *Biometrika*, 90, 4 (2003), 953–966. (cited on page 104)

Cunningham, R.; Herzog, T.; and London, R., 2012. *Models for Quantifying Risk*. ACTEX Academic series. Actex Publications. ISBN 9781566989336. (cited on page 56)

Currie, I. D.; Durban, M.; and Eilers, P. H., 2004. Smoothing and forecasting mortality rates. *Statistical Modelling*, 4, 4 (2004), 279–298. (cited on page 64)

Davies, P. L., 1987. Asymptotic behaviour of s-estimates of multivariate location parameters and dispersion matrices. *The Annals of Statistics*, 15, 3 (1987), 1269–1292. (cited on page 11)

Devlin, S. J.; Gnanadesikan, R.; and Kettenring, J. R., 1981. Robust estimation of dispersion matrices and principal components. *Journal of the American Statistical Association*, 76, 374 (1981), 354–362. (cited on pages 4 and 11)

Donoho, D. L., 2000. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1, 2000 (2000), 32. (cited on pages 103 and 105)

Enchev, V.; Kleinow, T.; and Cairns, A. J. G., 2017. Multi-population mortality models: fitting, forecasting and comparisons. *Scandinavian Actuarial Journal*, 2017, 4 (2017), 319–342. (cited on page 134)

Fan, J. and Gijbels, I., 1996. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, vol. 66. CRC Press. (cited on pages 73 and 74)

Fan, J.; Li, Q.; and Wang, Y., 2017. Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79, 1 (2017), 247–265. (cited on page 12)

Fan, J.; Liao, Y.; and Mincheva, M., 2013. Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B*, 75, 4 (2013), 603–680. (cited on pages 5, 6, 7, 25, 148, 152, and 153)

Fan, J.; Sun, Q.; Zhou, W.-X.; and Zhu, Z., 2018a. *Principal Component*

Analysis for Big Data, 1–13. In Wiley StatsRef: Statistics Reference Online. ISBN 9781118445112. (cited on pages 1, 5, and 6)

Fan, J.; Wang, K.; Zhong, Y.; and Zhu, Z., 2018b. Robust high dimensional factor models with applications to statistical machine learning. *arXiv preprint arXiv:1808.03889*, (2018). (cited on pages 5 and 6)

Fan, J.; Wang, W.; and Zhong, Y., 2019. Robust covariance estimation for approximate factor models. *Journal of econometrics*, 208, 1 (2019), 5–22. (cited on page 11)

Friedman, J.; Hastie, T.; and Tibshirani, R., 2001. *The Elements of Statistical Learning*, vol. 1. Springer series in statistics New York, NY, USA:. (cited on page 73)

Higuera, C.; Gardiner, K. J.; and Cios, K. J., 2015. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one*, 10, 6 (2015), e0129126. (cited on pages xx, 4, 106, 127, 129, and 130)

Hollmann, F. W.; Mulder, T. J.; and Kallan, J. E., 1999. *Methodology & Assumptions for the Population Projections of the United States: 1999 to 2010*. US Department of Commerce, Bureau of the Census, Population Division .... (cited on pages 17 and 61)

Hörmann, S.; Kidziński, Ł.; and Hallin, M., 2015. Dynamic functional principal components. *J. R. Stat. Soc. B*, 77 (2015), 319–348. (cited on pages 8, 18, 31, and 32)

Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24, 6 (1933), 417. (cited on page 1)

Huang, J.; Horowitz, J. L.; and Wei, F., 2010. Variable selection in nonparametric additive models. *Ann. Statist.*, 38, 4 (08 2010), 2282–2313. (cited on page 1)

Hubert, M.; Rousseeuw, P. J.; and Vanden Branden, K., 2005. Robpca: a new approach to robust principal component analysis. *Technometrics*, 47, 1 (2005), 64–79. (cited on page 12)

Hyndman, R. J.; Ahmed, R. A.; Athanasopoulos, G.; and Shang, H. L., 2011. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55, 9 (2011), 2579–2589. (cited on page 134)

Hyndman, R. J.; Booth, H.; and Yasmeen, F., 2013. Coherent mortality forecasting: the product-ratio method with functional time series models. *Demography*, 50, 1 (2013), 261–283. (cited on page 1)

Hyndman, R. J. and Ullah, M. S., 2007. Robust forecasting of mortality and fertility rates: a functional data approach. *Computational Statistics & Data Analysis*, 51, 10 (2007), 4942–4956. (cited on pages 10, 64, 88, and 90)

Johnstone, I. M., 2001. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of statistics*, 29, 2 (2001), 295–327. (cited on pages 105 and 118)

Johnstone, I. M. and Titterington, D. M., 2009. Statistical challenges of high-dimensional data. (cited on page 103)

Jolliffe, I. T., 2002. *Principal component analysis*. Springer Series in Statistics. Springer-Verlag, New York, second edn. ISBN 0-387-95442-2. (cited on pages 1, 11, 18, 104, and 123)

Lam, C. and Yao, Q., 2012. Factor modeling for high-dimensional time series: inference for the number of factors. *The Annals of Statistics*, 40, 2 (2012), 694–726. (cited on pages 8, 21, 22, 23, 25, 30, 34, 61, and 145)

Lam, C.; Yao, Q.; and Bathia, N., 2011. Estimation of latent factors for high-dimensional time series. *Biometrika*, 98, 4 (2011), 901–918. (cited on pages 6, 8, 18, 21, 31, 32, 105, 118, 137, 145, and 149)

Lee, R. and Miller, T., 2001. Evaluating the performance of the lee-carter method for forecasting mortality. *Demography*, 38, 4 (2001), 537–549. (cited on page 64)

Lee, R. D. and Carter, L. R., 1992. Modeling and forecasting US mortality. *Journal of the American Statistical Association*, 87, 419 (1992), 659–671. (cited on pages 3, 7, 9, 17, 28, 30, 43, 45, 61, 67, 72, and 78)

Lee, S.; Zou, F.; and Wright, F. A., 2014. Convergence of sample eigenvalues, eigenvectors, and principal component scores for ultra-high dimensional data. *Biometrika*, 101, 2 (2014), 484–490. (cited on pages 103, 105, and 118)

Li, G. and Chen, Z., 1985. Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and monte carlo. *Journal of the American Statistical Association*, 80, 391 (1985), 759–766. (cited on pages 12 and 104)

Li, H. and O'Hare, C., 2017. Semi-parametric extensions of the Cairns-Blake-Dowd model: A one-dimensional kernel smoothing approach. *Insurance: Mathematics and Economics*, 77 (2017), 166 – 176. (cited on pages 63 and 73)

Li, H.; O'Hare, C.; and Zhang, X., 2015. A semiparametric panel approach to mortality modeling. *Insurance: Mathematics and Economics*, 61 (2015), 264 – 270. (cited on pages 63, 64, and 73)

Li, N. and Lee, R., 2005. Coherent mortality forecasts for a group of populations: An extension of the lee-carter method. *Demography*, 42, 3 (2005), 575–594. (cited on page 134)

Li, S.-H. and Chan, W.-S., 2005. Outlier analysis and mortality forecasting: the United Kingdom and Scandinavian countries. *Scandinavian Actuarial Journal*, 2005, 3 (2005), 187–211. (cited on pages 10 and 62)

Locantore, N.; Marron, J.; Simpson, D.; Tripoli, N.; Zhang, J.; Cohen, K.; Boente, G.; Fraiman, R.; Brumback, B.; Croux, C.; et al., 1999. Robust principal component analysis for functional data. *Test*, 8, 1 (1999), 1–73. (cited on page 13)

Lundström, H. and Qvist, J., 2004. Mortality forecasting and trend shifts: an application of the Lee-Carter model to Swedish mortality data. *International Statistical Review*, 72, 1 (2004), 37–50. (cited on page 62)

Maronna, R. A.; Martin, R. D.; Yohai, V. J.; and Salibián-Barrera, M., 2019. *Robust statistics: theory and methods (with R)*. John Wiley & Sons. (cited on page 11)

Martinussen, T. and Scheike, T. H., 2000. A nonparametric dynamic additive regression model for longitudinal data. *The Annals of Statistics*, 28, 4 (2000), 1000–1025. (cited on page 7)

McCarthy, D. and Mitchell, O. S., 2001. *Assessing the impact of mortality assumptions on annuity valuation: Cross-country evidence*. Pension Research Council, the Wharton School, University of Pennsylvania. (cited on page 56)

Millossovich, P.; Villegas, A. M.; and Kaishev, V. K., 2018. Stmomo: An r package for stochastic mortality modelling. *Journal of Statistical Software*, 84, 3 (2018). (cited on page 9)

Minsker, S., 2018. Sub-gaussian estimators of the mean of a random matrix with heavy-tailed entries. *The Annals of Statistics*, 46, 6A (2018), 2871–2903. (cited on page 12)

Morales-Jimenez, D.; Johnstone, I. M.; McKay, M. R.; and Yang, J., 2018. Asymptotics of eigenstructure of sample correlation matrices for high-dimensional spiked models. *arXiv preprint arXiv:1810.10214*, (2018). (cited on page 103)

Park, B. U.; Mammen, E.; Härdle, W.; and Borak, S., 2009. Time series modelling with semiparametric factor dynamics. *Journal of the American Statistical Association*, 104, 485 (2009), 284–298. (cited on page 66)

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 (1901), 559–572. (cited on page 1)

Pena, D. and Box, G. E., 1987. Identifying a simplifying structure in time series. *Journal of the American Statistical Association*, 82, 399 (1987), 836–843. (cited on page 61)

Peña, D. and Yohai, V. J., 2016. Generalized dynamic principal components. *Journal of the American Statistical Association*, 111, 515 (2016), 1121–1131. (cited on page 8)

Reiss, M. and Wahl, M., 2020. Nonasymptotic upper bounds for the reconstruction error of pca. *Annals of Statistics*, 48, 2 (2020), 1098–1123. (cited on page 134)

Renshaw, A. and Haberman, S., 2003. Lee–carter mortality forecasting: A parallel generalized linear modelling approach for england and wales mortality projections. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 52, 1 (2003), 119–137. (cited on pages 10 and 62)

Richman, R. and Wuthrich, M. V., 2019. A neural network extension of the lee-carter model to multiple populations. *Annals of Actuarial Science*, (2019), 1–21. (cited on page 10)

Rousseeuw, P. J. and Croux, C., 1993. Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88, 424 (1993), 1273–1283. (cited on page 12)

Shang, H. L. and Haberman, S., 2020. Forecasting multiple functional time series in a group structure: an application to mortality. *Astin Bulletin*, 50, 2 (2020), 357–379. (cited on pages 10 and 134)

She, Y.; Li, S.; and Wu, D., 2016. Robust orthogonal complement principal component analysis. *Journal of the American Statistical Association*, 111, 514 (2016), 763–771. (cited on pages 11 and 104)

Social Security Administration, 2019. Period life tables. [https://www.ssa.gov/oact/HistEst/PerLifeTables/2019/PerLifeTables2019.html; accessed 5-June-2019]. (cited on page 15)

Stock, J. H. and Watson, M. W., 2002. Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97, 460 (2002), 1167–1179. (cited on pages 5 and 61)

Stone, M., 1977. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, (1977), 44–47. (cited on page 72)

Su, L. and Wang, X., 2017. On time-varying factor models: estimation and testing. *Journal of Econometrics*, 198, 1 (2017), 84–101. (cited on pages 3, 63, 66, 67, and 71)

The Board of Trustees of the Federal OASDI Trust Funds, 2019. *The 2019 annual report of the Board of Trustees of the Federal Old-Age and*

Survivors Insurance and Federal Disability Insurance Trust Funds. U.S. government publishing office. (cited on pages 54 and 55)

Tuljapurkar, S.; Li, N.; and Boe, C., 2000. A universal pattern of mortality decline in the G7 countries. *Nature*, 405, 6788 (2000), 789. (cited on page 62)

University of California, Berkeley (USA) and Max Planck Institute for Demographic Research (Germany). *Human Mortality Database*. www.mortality.org [Accessed: 2018.07.10]. (cited on pages 9, 16, 42, and 77)

Vidal, R.; Ma, Y.; and Sastry, S., 2016. *Generalized Principal Component Analysis*. Springer. (cited on pages 104, 105, 107, and 111)

Wang, W. and Fan, J., 2017. Asymptotics of empirical eigenstructure for high dimensional spiked covariance. *Annals of statistics*, 45, 3 (2017), 1342. (cited on pages 105, 118, and 119)

Warshawsky, M., 1988. Private annuity markets in the united states: 1919-1984. *Journal of Risk and Insurance*, (1988), 518–528. (cited on page 56)

Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; and Ma, Y., 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. (2009), 2080–2088. (cited on page 12)

Yang, S. S.; Yue, J. C.; and Huang, H.-C., 2010. Modeling longevity risks using a principal component approach: A comparison with existing stochastic mortality models. *Insurance: Mathematics and Economics*, 46, 1 (2010), 254–270. (cited on pages 10 and 62)

Zhao, Q.; Meng, D.; Xu, Z.; Zuo, W.; and Zhang, L., 2014. Robust principal component analysis with complex noise. In *International conference on machine learning*, 55–63. (cited on page 13)