Advances in Complex Data Analysis

By

CHENCHENG CAI

A dissertation submitted to the
School of Graduate Studies
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Statistics
Written under the direction of
Rong Chen
and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2020

ABSTRACT OF THE DISSERTATION

Advances in Complex Data Analysis

by CHENCHENG CAI

Dissertation Director: Rong Chen

With the increasing availability of big data, it is challenging to analyze complex data that is high dimensional; high volume but heterogeneous; or imposed with constraints. My dissertation compiles researches from three different areas to address the solutions to those challenging problems in complex data analysis.

In Part I, we first solve the constrained sampling problem in state space models, which is usually difficult due to potential strong constraints (Lin et al., 2010). The proposed Sequential Monte Carlo with constraints (SMCc) algorithm provides a general framework to sample efficiently from a state space model with constraints. An optimal priority score used in the resampling step of sequential Monte Carlo (SMC) is introduced as a compromise between accuracy and computation. Several computationally efficient ways of approximating the optimal priority are presented.

The second half of Part I focuses on utilizing state space models and SMC to solve high dimensional optimization problems, in which traditional optimization algorithms usually have their limitations. We propose to first reformulate the optimization problem into the likelihood function of an artificially designed state space model (the emulation step) and then find the optimal solution through a novel simulated annealing algorithm for state space models (the annealed SMC step). The procedure is demonstrated with several canonical statistical examples.

In Part II, we propose an individualized group learning (iGroup) framework, lying at the intersection fusion learning and individualized inference, to provide a more concrete statistical inference on a particular individual of interest, by aggregating information of

similar individuals from a potentially heterogeneous population. The optimality of such a methodology is shown under the asymptotic setting that the population size approaches infinity while each individual has a finite number of observations. The improvement of *i*Group over individual level estimate and the population level estimate (as in traditional fusion learning) are demonstrated with simulations and real data examples.

In Part III, we consider the family of KoPA approaches, which approximate a high dimensional matrix with one or more Kronecker products. Using Kronecker product instead of vector outer product introduces much higher flexibility in choosing the configuration (sizes of the two smaller matrices), while it gives rise to the problem of choosing the optimal one. An extended information criterion is proposed to automatically select the optimal configuration. Consistency of the configuration selection is provided with rigorous analysis. In addition, the KoPA approach can be extended to matrix completion problems as well with a superior performance over traditional SVD as demonstrated in Part III with a real image example.

Acknowledgement

Acquiring a PhD degree has been a truly challenging and life-changing experience for me and it would not have been possible to do without the support and advisory from many people.

Firstly, I would like to express my deepest gratitude to my advisor, Professor Rong Chen for his continuous help, endless guidance and generous support. I still remember the day when I sat in his office when he asked me why not consider pursuing a PhD degree when I was a master student. Starting from then, I took the opportunity to devote myself in statistical research. As time passed, his patience, motivation and encouragement help me through all my time of research. I am especially indebted for the flexibility he gave to determine research of interests, from which I have great experience in diverse research areas, as well as chances to cooperate with others. Also, his brilliant ideas and deep insights shared with me always significantly inspire and motivate my research. Prof. Chen is more than an advisor to me, his devotion and passion in statistics deeply touched me. And thus, I decide to pursue an academic career in statistics and hope there is one day I can become a professor, just like him. I cannot be more fortunate to have Prof. Chen as my advisor.

Secondly, I am also extremely grateful to Professor Ming-ge Xie, Professor Han Xiao and Professor Ming Lin for their constructive advice and kind support on various aspects of my projects. Without their expertise, inspiration, and encouragement, I cannot achieve such diversity in research topics. I enjoy so much working with them and learn a lot from cooperating with them.

Of course, my PhD degree would not have been possible without the financial support of the Department of Statistics, Rutgers University. I wish to appreciate all the support and help from the faculty, staff, and colleague in the department. I especially grateful

to the department chair, Professor Regina Liu, who provided me great resources for my future academic job searching. Moreover, I would like to say thanks to the former graduate director, Professor John Kolassa and the current graduate directors, Professor Harry Crane and Professor Tirthankar DasGupta for their continuous support and guidance.

Last but not the least, I would like to thank my parents for their unconditional love and support, as well as my girlfriend for her encouragement and help.

Dedication

To my parents Zhixin and Weihong; To my girlfriend Yimeng

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

With the massive data readily available in the era of big data, advanced statistical methodologies for analyzing complex data are in high demand. Big data is often characterized by high volume, high velocity and high variety. New challenges arise when analyzing such data. For example, (a) in the area of Monte Carlo methods, it is usually difficult to draw samples efficiently from a high dimensional distribution with constraints (Durham and Gallant, 2002; Lin et al., 2010). (b) In the area of optimization, some traditional algorithms (Bertsekas, 1997; Anandkumar et al., 2014; Arora et al., 2012) does not perform well on high dimensional optimization problems because it is either difficult to estimate a high dimensional gradient or too expensive to search the parameter space exhaustively. (c) In fusion learning studies where results from different sources are combined to make a coherent conclusion, it is challenging to determine what to combine and how to combine when the population is potentially heterogeneity and when only one study instead of the overall population is particular interest (Shen et al., 2019). (d) In the field of high dimensional data analysis, studies have been focusing on replacing the low rank singular value decomposition (SVD) with the Kronecker product in certain engineering problems (Werner et al., 2008; Duarte and Baraniuk, 2012; Kamm and Nagy, 1998). The problem of the determining the dimensions of the two smaller matrices that compose the Kronecker product in an automatic way remains open and needs theoretical foundation. There also exist tons of other challenges involving data quantity and quality in addition to the aforementioned examples.

In this dissertation, researches on three areas of study will provide the answers to the aforementioned four challenges. Specifically, Part I focuses on the State Space Model and

the Sequential Monte Carlo methods. A Sequential Monte Carlo with constraints (SMCc) algorithm is proposed to draw samples efficiently from a constrained state space model. The general principle of state space emulation will be discussed to reformulate high dimensional optimization problems to most likely path problems within state space model frameworks. A novel, SMC-based simulated annealing approach will be presented to solve the most likely path problems in state space model numerically.

In Part II, we propose the framework of *i*Group, short for individualized group learning, to address the individualized inference problem within a potential heterogeneous population with a modified fusion learning approach. Theoretical guarantee of optimality is also provided.

Part III introduces a family of Kronecker otimes Product Approximation (KoPA) methods that aims to approximate high dimensional matrix with one or more Kronecker products. In addition to the flexibility brought by the Kronecker product, we provide a theoretical foundation of consistent configuration estimation to the KoPA model.

1.1 Motivation

1.1.1 Part I: Sequential Monte Carlo

Sequential Monte Carlo (SMC) is a class of Monte Carlo methods that is often widely used to draw samples efficiently and sequentially from a state space model, which describes the dynamics of a sequence of observations with the help of an additional sequence of latent variables. SMC along with state space models is widely used in statistics, economics and engineering to make statistical inference in a numerical way, when the dynamics or the likelihood functions are high-dimensional and are infeasible.

The traditional SMC approaches aim to sample from a state space model with frequent observations (Doucet et al., 2001; Liu and Chen, 1998). Increasing amount of constrained problems require for a more efficient SMC algorithm to draw samples from the constrained state space models as the traditional SMC does perform well with strong constraints. See Lin et al. (2010) for the diffusion bridge problem where traditional SMC algorithms have their limitations.

Motivated by the needs for a general SMC framework for the constrained sampling problem, we proposed a new SMC framework named "SMCc". In SMCc, we formulate the constrained sampling problem and provide a general framework to sample from the constrained systems with a specifically designed resampling strategy. An optimal priority score used in the resampling step is involved and three different approaches to approximate it through either parametric method or sampling method.

In addition to the constrained sampling problem, we notice that many high dimensional optimization problems are equivalent to state space models after certain re-phasing of likelihood functions. The gradient descent based optimization algorithms often suffer from an inaccurate estimation of gradient when the dimension is high. Non-convex optimization algorithms usually turns out to be an exhaustive search over the space, which results in a huge computational cost for high dimensional problems. Therefore, in the second half of Part I, we consider the reverse: in the first step, an optimization problem is transformed into the most likely path problem of an artificially designed state space model; in the second step, an annealing algorithm based on SMC is proposed to find the most likely path numerically. We call the first step state space emulation and name the algorithm in second step "annealed SMC".

1.1.2 Part II: Individualized Group Learning

It is widely recognized that aggregate information from different sources or independent studies helps to provide a stronger conclusion as more data is fused together. Meta analysis or fusion learning is the area of research in statistics that investigate the optimal way of data fusion (Chen and Xie, 2014; Liu et al., 2014, 2015; Yang et al., 2016). In the fusion learning, it is often assumed that the parameter of interest from different studies/sources are exactly the same. The assumption is usually satisfied when the objectives of the studies are the same or when each source is a subset of data from the same database. However, if the population of studies is heterogeneous, that is, different studies/sources may potentially have different parameters of interest, the aggregation may yield invalid results. In addition, when the population is heterogeneous, one is often more interest in the parameter of a particular individual in the population instead of the population-level average such as

in precision medicine (Insel, 2009; Hamburg and Collins, 2010; Qian and Murphy, 2011) and individualized inference (Liu and Meng, 2016). Motivated by the potential population heterogeneity in fusion learning problems and the increasing need for individualized inference, Part II focuses on a new fusion learning based individualized inference approach called *i*Group. Particularly, *i*Group aims to make an individualized inference for a target individual of interest by only aggregating similar individuals. A previous work Shen et al. (2019) aims to provide the individualized fusion learning approach with the same goal under the asymptotic setting that the population size is limited but the number of observations for each individual approaches infinity. *i*Group considers a different asymptotic setting, which assumes the population size approaches infinity while each individual has a limited number of observations. The bias problem is more significant under the *i*Group setting. A complete theoretical analysis is provided to show that the proposed *i*Group method is optimal under the corresponding asymptotic setting.

1.1.3 Part III: Automatic Kronecker Product Approximation

With the increasing availability of high dimensional matrix/tensor data, it is a challenge to store, represent and model such data. Sparsity is often assumed for high dimensional matrices such that the data matrix may have a limited number of non-zero entries (sparsity in observations), or have limited number of non-zero Fourier components (sparsity in frequencies), or have a limited number of non-zero singular values (sparsity in the spectrum). Especially the low rank assumption (sparsity in the spectrum) is pretty common as in factor analysis where the covariance matrix is represented with a sum of several rank-1 matrices, each of which is an outer product of two vectors.

Instead of low rank assumptions, we observe that the Kronecker product is another mathematical construction of large matrices in statistics and engineering. The dimensions of the two smaller matrices in the Kronecker product is called the configuration. It has been investigated to approximate a large matrix with given configuration (Van Loan and Pitsianis, 1993). For a given matrix, there usually exists more than one way to represent it with Kronecker products, corresponding to different configurations. However, it remains open to determine the configuration when it is unknown. Motivated by the flexibility of

Kronecker product approximation and its potential applications in a variety of research areas, in Part III, we focus on approximating a high-dimensional matrix with one or more Kronecker products as in (1.1).

$$\boldsymbol{Y} \approx \sum_{k=1}^{K} \boldsymbol{A}_k \otimes \boldsymbol{B}_k, \tag{1.1}$$

where $\otimes$ stands for the Kronecker product, and the configuration of each Kronecker product is unknown.

The (original) KoPA assumes the k-components in (1.1) have the same unknown configuration. We show that the configuration can be determined through an extended version of information criteria. Theoretical foundation is established the consistency of configuration determination in KoPA problems.

One extension is the hybrid KoPA (hKoPA), where the configurations for the K terms in (1.1) can be different. We provide an alternating least square algorithm to estimate $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ when the configurations are known and a greedy iterative algorithm to estimate $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ as well as their configuration when the configurations are unknown.

Another extension is the matrix completion problem (MC-KoPA), where only partial entries of $\boldsymbol{Y}$ are observed, under the same assumption of the original KoPA model. In MC-KoPA, the unknown common configuration can be determined through a modified version of the information criteria that was used in KoPA model.

All the original KoPA model and its variations are presented with simulations and real examples in image processing.

1.2 Dissertation Outline

This dissertation is divided into three parts according to the topic of research: part I: Sequential Monte Carlo (SMC); part II: Individualized Group Learning (*i*Group); part III: Automated Kronecker Product Approximation (KoPA). The hierarchical structure of chapters is depicted in Figure 1.1.

Part I consists of four chapters. Chapter 2 provides a review on state space models and the sequential Monte Carlo framework. In Chapter 3, the constrained sampling problems

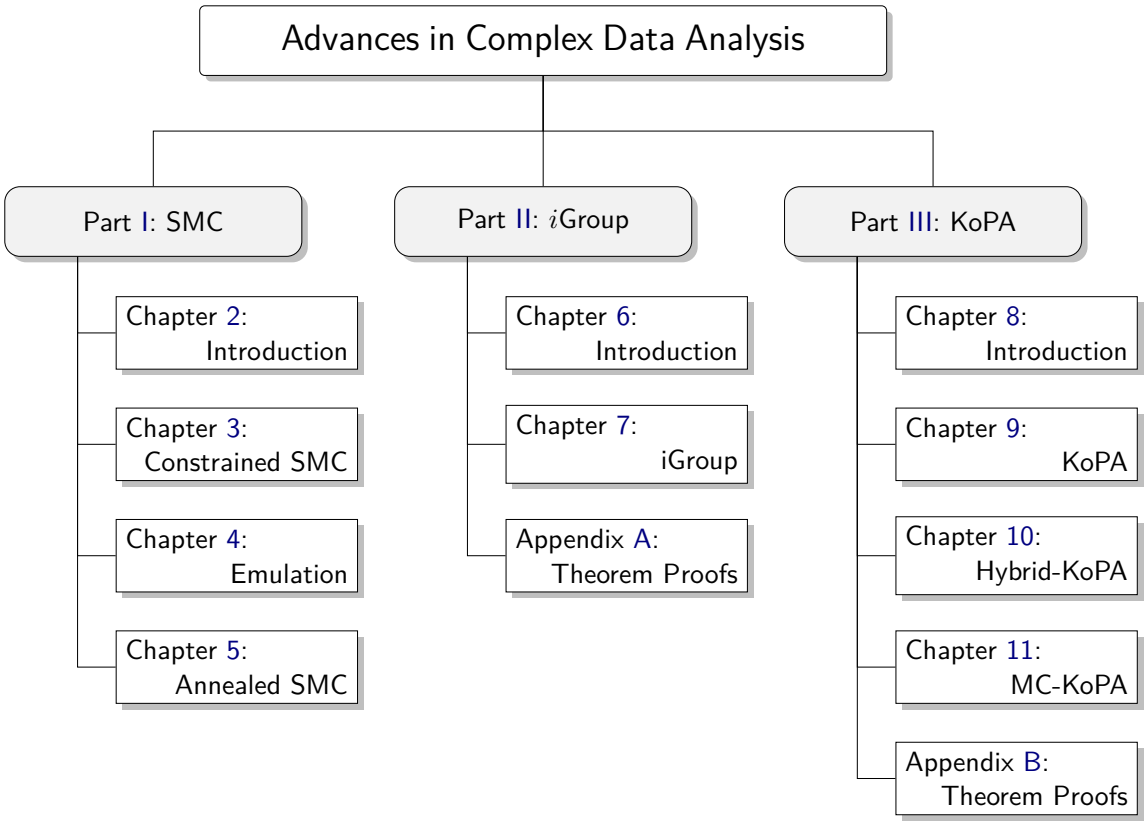


Figure 1.1: Hierarchical Structure of Thesis.

are discussed and the proposed algorithm SMCc is demonstrated with simulations and examples. Chapter 4 provides the details of state space emulation with several examples and Chapter 5 introduces the annealed SMC algorithm that optimizes the emulated state space models resulting from Chapter 4.

Part II four chapters in the main content. Chapter 6 reviews the history of fusion learning and individualized inference. Chapter 7 gives the main framework of individualized group learning. Detailed theoretical analysis is represented in Section 7.2 with the corresponding rigorous proofs in Appendix A. Varies of simulations and real data examples of *i*Group are demonstrated in Section 7.3.

Part III discusses the family of KoPA approaches. The preliminary knowledge of Kronecker product is introduced in Chapter 8. The methodology, theoretical analysis and examples of KoPA model are given in Chapter 9. Two extensions, the hybrid-KoPA and the KoPA for matrix completion problems are introduced in Chapter 10 and Chapter 11. Theorem proofs are given in Appendix B.

PART I

Sequential Monte Carlo for Constrained Problems and High Dimensional Optimizations

CHAPTER 2

Preliminary: Sequential Monte Carlo

2.1 State Space Model

2.1.1 Stochastic Dynamic Systems

Stochastic dynamic systems are often used to model the dynamic behavior of random variables with a wide range of applications in physics, finance, engineering and other fields. Denote the random variable of interest as x_t, where the subscript t is used to emphasize the time dependence. As t is allowed to take any value of non-negative real numbers, the dynamics of x_t is often modeled by a stochastic differential equation (SDE). For example, in mathematical finance, the stock price X_t is assumed to follow a geometric Brownian motion (Hull, 2015)

$$dx_t = \mu x_t dt + \sigma x_t dW_t, \tag{2.1}$$

where μ is the drift, σ is the volatility and W_t is a standard Brownian motion.

Although the stochastic differential equations as in (2.1) provide a precise description, one is often interested in the discretized version of the SDE. On the one hand, from the perspective of simulation and numerical analysis, it is impractical to simulate the complete process $\{x_t : 0 \leqslant t \leqslant T\}$ at every t. Instead, it is a common practice to simulate the discrete path $\{x_t : t = 0, \delta, 2\delta, \ldots, K\delta = T\}$ with a time resolution δ and let $\delta \to 0$ to approximate the continuous-time process. On the other hand, the stochastic process data are usually collected/observed at only a finite/countable number of time points. For example, the historical stock price is usually recorded on a daily base, and the national gross domestic

product (GDP) data is calculated every quarter. In this circumstance, it is more reasonable to model them as discrete-time stochastic processes.

Without loss of generality, we assume the data collection or observation occurs in a periodic manner such that the corresponding times are $t = k\delta$ for $k = 0, 1, \ldots$ and $\delta > 0$. To ease our notation, we use δ as one unit of time and the time points of observation can be simply to $t = 0, 1, 2, \ldots$.

With above time notations, the SDE for the stock price in (2.1) can be rewritten as

$$\log x_{t+1} = \log x_t + \left[\mu + \frac{1}{2}\sigma^2\right]\delta + \sigma\epsilon_{t+1},$$

where δ is the time step and $\epsilon_{t+1} \sim N(0, \delta)$.

2.1.2 State Space Model

State space model (SSM) is a widely-used discrete-time stochastic model (Doucet et al., 2001; West and Harrison, 1998; Liu and Chen, 1998). Specifically, in the state space model, we assume that there exists a sequence of latent random variables, whose dynamics are governed by an initial state distribution $f_0(x_0)$ and a forward propagation distribution $f_t(x_t \mid x_{0:t-1})$, which is often called the "state equations". In addition, we assume at each time t, a random variable y_t is generated independently according to the conditional distribution $g_t(y_t \mid x_t)$. In many applications, $y_{1:T} = (y_1, \ldots, y_T)$ is interpreted as observations. Hence, the conditional distribution g_t is often referred as the "observation equations".

We summarize the dynamics of the state space model in (2.2) and (2.3).

$$\text{state equation:} \qquad p(x_t \mid x_{0:t-1}) = f_t(x_t \mid x_{0:t-1}), \tag{2.2}$$

$$\text{observation equation:} \qquad p(y_t \mid x_t) = g_t(y_t \mid x_t), \tag{2.3}$$

where $p(\cdot)$ is the general notation for probabilities.

The state equation $f_t(\cdot)$ in (2.2) and the observation distribution $g_t(\cdot)$ in (2.3) are often assumed to be known exactly or up to an unknown parameter of interest θ. Especially when the state space model is a discretized version of a continuous-time stochastic process,

the state equation $f_t(\cdot)$ can be obtained from the stochastic differential equation of the stochastic process. For instance, the state equation for the stock price example in Section 2.1.1 is known except the parameter $\theta = (\mu, \sigma^2)$.

The dynamics of (2.2) and (2.3) reveals the dependence structure of the state space model. On the one hand, the distribution of x_t only depends on all previous states $x_{0:t-1}$, which preserves causality of the underlying latent stochastic process $\{x_t : t \in \mathbb{Z}_+\}$. On the other hand, the observations $y_{1:T}$ are conditionally independent given the values of latent variables $x_{1:T}$. For a better demonstration of the dependence between the variables, we demonstrate the diagram in Figure 2.1, where each arrow represents a conditional distribution and one arrow is independent from the other.

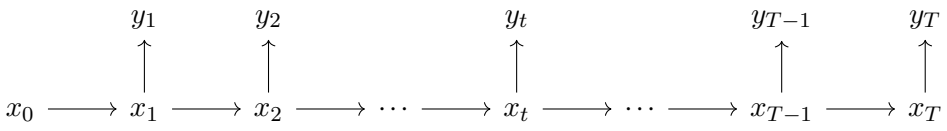


Figure 2.1: Diagram of State Space Model

When the distribution of x_t depends only on x_{t-1} such that $p(x_t \mid x_{0:t-1}) = f_t(x_t \mid x_{t-1})$, the system is called Markovian. When both state equations and observation equations are time-independent, the system is time-homogeneous. If a state space model is Markovian and time-homogeneous, and in addition x_t takes values in a common support of finite elements, such a model is often called "hidden Markov model" (HMM) (Stratonovich, 1965; Baum and Petrie, 1966). The hidden Markov model has been extensively investigated in statistics, computer science and engineering (Juang and Rabiner, 1991; Bishop and Thompson, 1986; Ghahramani and Jordan, 1996; Baum and Eagon, 1967). Compared to hidden Markov model, the state space model has a more general setting, which allows for continuous latent variables x_t and allows for non-Markovian processes.

2.1.3 Statistical Inference for State Space Models

When the state and observation equations are known up to an unknown parameter of interest θ, we denote them by $f_t(x_t \mid x_{0:t-1}; \theta)$ and $g_t(y_t \mid x_t; \theta)$ to emphasize the dependence.

The joint probability of $x_{0:T}$ and $y_{1:T}$ is given by

$$p(x_{0:T}, y_{1:T} \mid \theta) = f_0(x_0;\theta) \prod_{t=1}^{T} f_t(x_t \mid x_{0:t-1};\theta) g_t(y_t \mid x_t;\theta).$$

Then the likelihood function of θ given the observations $y_{1:T}$ can be obtained by integrating out the latent variables $x_{0:T}$ such that

$$L(\theta) = p(y_{1:T} \mid \theta) = \int f_0(x_0;\theta) \prod_{t=1}^{T} f_t(x_t \mid x_{0:t-1};\theta) g_t(y_t \mid x_t;\theta) dx_{0:T}. \tag{2.4}$$

A maximum likelihood estimator of θ can be calculated by maximizing $L(\theta)$.

When the sequence of latent variables $x_{0:T}$ are of primary interest, the systems can be interpreted under Bayesian framework. Specifically, we assume the state and observation equations are exactly known. The following distribution of $x_{0:T}$, which can be measured before observation $y_{1:T}$, is viewed as the prior distribution of $x_{0:T}$.

$$\pi(x_{0:T}) = f_0(x_0) \prod_{t=1}^{T} f_t(x_t \mid x_{0:t-1}). \tag{2.5}$$

The corresponding likelihood function is given by

$$p(y_{1:T} \mid x_{0:T}) = \prod_{t=1}^{T} g_t(y_t \mid x_t). \tag{2.6}$$

Therefore, the posterior of $x_{1:T}$ given $y_{1:T}$ is

$$\pi(x_{0:T} \mid y_{1:T}) \propto f_0(x_0) \prod_{t=1}^{T} f_t(x_t \mid x_{0:t-1}) g_t(y_t \mid x_t). \tag{2.7}$$

Note that the likelihood function in (2.6) is different from the one in (2.4) because the parameters of interest are different in these two problems.

Maximizing the posterior in (2.7) gives the most likely values of $x_{0:T}$, which is known as the most likely path (MLP) problem. Such an inference problem of the latent variables is common in hidden Markov models. For example, in speech recognition (Juang and Rabiner, 1991), the goal is to recover the true words (x_t's) given the audio data $y_{1:T}$.

The most likely path problem can be viewed as a special case of the smoothing problem, where one is particularly interested in the conditional distribution $p(x_s \mid y_{1:t})$ for some $s < t$. In contrast to the smoothing problem, where most current observations $y_{1:t}$ are used to estimate a past state x_s, $s < t$, the filtering problem uses the same information to estimate the most current state, that is $p(x_t \mid y_{1:t})$.

The filtering problem is closely related to the estimation of the likelihood function (2.4). It can be seen from the following relationship between the filtering distribution $p(x_t \mid y_{1:t})$ and the conditional likelihood $p(y_{t+1} \mid y_{1:t}, \theta)$.

$$p(y_{t+1} \mid y_{1:t}, \theta) = \iint p(x_t \mid y_{1:t}; \theta) f_{t+1}(x_{t+1} \mid x_t; \theta) g_{t+1}(y_{t+1} \mid x_{t+1}; \theta) dx_t dx_{t+1}.$$

2.2 Sequential Monte Carlo Framework

Sequential Monte Carlo (SMC) is a class of numeric algorithms belonging to the family of Monte Carlo methodology. Example of SMC literature and applications includes Kong et al. (1994); Avitzour (1995); Liu and Chen (1995); Kitagawa (1996); Kim et al. (1998); Pitt and Shephard (1999); Chen et al. (2000); Doucet et al. (2001); Fong et al. (2002); Godsill et al. (2004) among many others. Sequential Monte Carlo methods conduct statistical inference based on a set of Monte Carlo samples which are drawn efficiently in a sequential fashion from the posterior distribution in (2.7) of a state space model. The technical details of varies sequential Monte Carlo algorithms will be reviewed in this section. The core idea of SMC algorithms is to generate Monte Carlo samples $\{x_{0:t}^{(j)}\}_{j=1}^{n}$ from those at the previous time stamp $\{x_{0:t-1}^{(j)}\}_{j=1}^{n}$ such that the set of Monte Carlo samples grows sequentially as the time evolves.

Sequential Monte Carlo has an old name "particle filter", which is misleading in the way that the word "filter" gives the impression of solving filtering problems (see Section 2.1.3). Although SMC or particle filter is a powerful tool in dealing with the filtering needs, the methodology is also applied to the smoothing and other more advanced problems in modern applications, which will be discussed in this chapter.

2.2.1 Principle of Importance Sampling

In Monte Carlo methods, if the parameter of interest θ can be written as the expectation of a random variable $h(x)$ under a probability measure μ on x such that

$$\theta = \mathbb{E}_\mu[h(x)] = \int h(x) d\mu(x) < \infty,$$

then a Monte Carlo estimator of θ can be obtained by the sample average

$$\hat{\theta}_N = N^{-1} \sum_{j=1}^{N} h(x^{(j)}),$$

where $\{x^{(j)}\}_{j=1}^N$ are drawn i.i.d. from the probability measure μ. By the central limit theorem, the asymptotic distribution of the Monte Carlo estimator is

$$\sqrt{n}(\hat{\theta}_n - \theta) \xrightarrow{\mathcal{D}} \mathcal{N}(0, V), \tag{2.8}$$

where $V = \int h^2(x) d\mu(x) - \theta^2$ assuming $h^2(x)$ integrable under μ.

In importance sampling, the distribution μ is called the target distribution and the samples $\{x^{(j)}\}_{j=1}^N$ can be drawn from another distribution $\nu(x)$, which is known as the sampling distribution or proposal distribution. In addition, each sample $x^{(j)}$ is equipped with a non-negative weight $w^{(j)} = d\mu(x^{(j)})/d\nu(x^{(j)})$, which is the Radon-Nikodym derivative between the target and the proposal probability measures. When μ and ν are continuous measures with corresponding densities f_μ and f_ν, the weight function is simply $w(x) = f_\mu(x)/f_\nu(x)$, which requires $f_\nu(x)$ to be 0 whenever $f_\mu(x)$ is 0. The condition is often expressed as "μ is absolutely continuous with respect to ν".

The weighted sample set $\{(x^{(j)}, w^{(j)})\}_{j=1}^N$ is said to be properly weighted to the distribution μ if for any measurable function $h(\cdot)$ we have

$$\tilde{\theta}_N = \frac{\sum_{j=1}^N w^{(j)} h(x^{(j)})}{\sum_{j=1}^N w^{(j)}} \xrightarrow{P} \mathbb{E}_\mu[h(x)] = \theta. \tag{2.9}$$

The Radon-Nikodym derivative weight function ensures (2.9) since

$$N^{-1}\sum_{j=1}^{N} w^{(j)} h(x^{(j)}) \xrightarrow{P} \int \frac{d\mu(x)}{d\nu(x)} h(x) d\nu(x) = \theta,$$
$$N^{-1}\sum_{j=1}^{N} w^{(j)} \xrightarrow{P} \int \frac{d\mu(x)}{d\nu(x)} d\nu(x) = 1,$$

by law of large numbers. Similar to (2.8), we have the asymptotic distribution for $\tilde{\theta}_n$ as

$$\sqrt{n}(\tilde{\theta}_n - \theta) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \tilde{V}), \tag{2.10}$$

where

$$\tilde{V} = \int \left(\frac{d\mu(x)}{d\nu(x)} h(x)\right)^2 d\nu(x) - \theta^2 \geqslant \frac{\left[\int |h(x)| d\mu(x)\right]^2}{\int d\nu(x)} - \theta^2 = \left[\int |h(x)| d\mu(x)\right]^2 - \theta^2,$$

where the inequality is a direct consequence of Cauchy-Schwartz inequality. By Jensen's inequality, we have $\tilde{V} \leqslant V$. The optimal variance is obtained when $d\nu(x)/d\mu(x) \propto |h(x)|$.

When the function $h(\cdot)$, whose expectation under μ is the estimand, is not specified, it is common to look into the χ^2 divergence between μ and ν. Specifically, the χ^2 divergence $D_{\chi^2}(\mu||\nu)$ is defined as

$$D_{\chi^2}(\mu||\nu) = \int \left[\left(\frac{d\mu(x)}{d\nu(x)}\right)^2 - 1\right] d\nu(x) = \int w^2(x) d\nu(x) - 1,$$

which is an increasing function of $\mathbb{E}_\nu[w^2] = \int w^2(x) d\nu(x)$. Consequently, the concept of effective sample size arises for finite sample sets. A finite sample estimate for $E_\nu[w^2]$ is $N^{-1}\sum_{j=1}^{N}\left[w^{(j)}\right]^2$, which is 1 when $w^{(j)} \equiv 1$ and greater than 1 otherwise. Accordingly, we define the effective sample size (ESS) as follows

$$\text{ESS} = \frac{\left[\sum_{j=1}^{N} w^{(j)}\right]^2}{\sum_{j=1}^{N}\left[w^{(j)}\right]^2}, \tag{2.11}$$

which takes the maximum at N when the weights are evenly distributed. A larger effective sample size indicates a smaller χ^2-divergence between the sampling distribution and the

target distribution.

As a summary, the benefit of importance sampling is twofold. On the one hand, when the target distribution $\mu(x)$ is difficult to sample from, sampling according to an arbitrary proposal distribution $\nu(x)$ is relatively easy. For instance, one can choose $\nu(x)$ to be Gaussian. On the other hand, by comparing the asymptotic variance of $\hat{\theta}_n$ and $\tilde{\theta}_n$ in (2.8) and (2.10), the constant weight is usually not optimal in estimating θ. By choosing the optimal proposal distribution, the asymptotic variance of the (weighted) mean estimator is reduced.

2.2.2 Sequential Importance Sampling

The principle of importance sampling can be utilized to draw sample trajectories $\{x_{0:T}^{(i)}\}_{i=1}^N$ from the posterior distribution (2.7) in a sequential way. Specifically, let $q_0(x_0), q_1(x_1 \mid x_0), \ldots, q_T(x_T \mid x_{0:T-1})$ be a sequence of proposal distributions of arbitrary choices as long as the target distribution (2.7) is absolutely continuous with respect to the joint proposal distribution $q_0(x_0)\prod_{t=1}^T q_t(x_t \mid x_{0:t-1})$. According to the principle of importance sampling, the proper weight at time T should be

$$\begin{aligned} w_T(x_{0:T}) &\propto \frac{f_0(x_0)\prod_{t=1}^T f_t(x_t \mid x_{0:t-1}) g_t(y_t \mid x_t)}{q_0(x_0)\prod_{t=1}^T q_t(x_t \mid x_{0:t-1})} \\ &= \frac{f_0(x_0)}{q_0(x_0)} \prod_{t=1}^T \frac{f_t(x_t \mid x_{0:t-1}) g_t(y_t \mid x_t)}{q_t(x_t \mid x_{0:t-1})}, \end{aligned} \tag{2.12}$$

which has a sequential structure with the weight increment term

$$\frac{f_t(x_t \mid x_{0:t-1}) g_t(y_t \mid x_t)}{q_t(x_t \mid x_{0:t-1})}.$$

Note that the relation in (2.12) is in a proportional form, where the value of the right-hand side is feasible. With rigorous calculation, the missing term in (2.12) is $1/p(y_{1:T})$, which is infeasible but constant for any $x_{0:T}$.

With both proposal distribution and weight function in a sequential form, the proposal sampling and weight adjustment can therefore be conducted sequentially at every time $t = 1, \ldots, T$ instead of deferring all weight calculation to time T. The procedure is known as sequential importance sampling (SIS), which is depicted in Algorithm 1.

Algorithm 1: Sequential Importance Sampling

Draw $\{x_0^{(i)}\}_{i=1}^N$ from $q_0(x_0)$;

Set $w_0^{(i)} \leftarrow 1$ for $i = 1, \ldots, N$;

for $t = 1, \ldots, T$ **do**

  for $i = 1, \ldots, N$ **do**

    Draw $x_t^{(i)}$ from $q_t(x_t \mid x_{0:t-1}^{(i)})$;

    Set

$$w_t^{(i)} \leftarrow w_{t-1}^{(i)} \frac{f_t(x_t^{(i)} \mid x_{0:t-1}^{(i)}) g_t(y_t \mid x_t^{(i)})}{q_t(x_t^{(i)} \mid x_{0:t-1}^{(i)})}; \tag{2.13}$$

  end

end

return $\{(x_{0:T}^{(i)}, w_T^{(i)})\}_{i=1}^N$.

Because of the proportion form in (2.12), the weight update step in (2.13) misses a constant term $p(y_t \mid y_{1:t-1})$ as well. This helps in estimating the likelihood function in the way that

$$\hat{p}(y_{s+1:t} \mid y_{0:s}) = \frac{\sum_{i=1}^N w_t^{(i)}}{\sum_{i=1}^N w_s^{(i)}},$$

provides a Monte Carlo estimate of the conditional likelihood $p(y_{s+1:t} \mid y_{0:s})$ with $0 < s < t$.

2.2.3 Sequential Importance Sampling with Resampling

A common issue encountered in the sequential importance sampling algorithm in Algorithm 1 is weight collapse. As an extreme example, when at time t the majority of weight is assigned to one sample $x_{0:t}^{(1)}$, then at time $t+1$, only one useful value for x_{t+1} is obtained — $x_{t+1}^{(1)}$, as all other samples have negligible weights. This weight collapse reduces the diversity and representativity of Monte Carlo samples.

Resampling alleviates this issue by reproducing samples with higher weights and eliminating samples with lower weights. For the aforementioned extreme example, one can reproduce the sample $x_{0:t}^{(1)}$ N times such that the new sample set $\{\tilde{x}_{0:t}^{(i)} = x_{0:t}^{(1)}\}_{i=1}^N$ are now assigned with equal weights. As a result, at time $t+1$, N new values of x_{t+1} are generated

verses only one value without reproduction.

The resampling step in sequential Monte Carlo embraces a stochastic algorithm. Suppose at time t, $\{(x_{0:t}^{(i)}, w_t^{(i)})\}_{i=1}^N$ is the properly-weighted sample set. A resampling step is redeemed by first generating $j_1, \ldots, j_N$ identically distributed to the probability

$$P[j_i = k] = \beta_k \quad \text{for } k = 1, \ldots, N, \tag{2.14}$$

and then assigning

$$\begin{aligned} x_{0:t}^{(i)} &\leftarrow x_{0:t}^{(j_i)}, \\ w_t^{(i)} &\leftarrow w_t^{(j_i)} / \beta_{j_i}, \end{aligned}$$

for all $i \in [N]$. The resampling probability in (2.14) employs the multinomial distribution with probabilities $(\beta_1, \ldots, \beta_N)$, which are known as the priority scores. The priority scores indicate the re-sampler's preference over different samples — samples with higher priority scores are more likely to be reproduced. A common choice of the priority score is the weight such that $\beta_k = w_t^{(k)}$. But the choice of priority score is arbitrary, and after resampling the weighted sample set remains properly-weighted.

The sequential importance sampling algorithm with the extra resampling step is known as sequential importance sampling with resampling (SISR), which is demonstrated in Algorithm 2.

The index set $\{j_1, \ldots, j_N\}$ for new samples in the resampling step can be generated with multinomial distribution random number generators. More advanced schemes to reduce variation introduced by the resampling step include residual resampling (Liu and Chen, 1998) and stratified resampling (Carpenter et al., 1999).

The resampling step is not necessary to each time t as it introduces extra variation. A fixed schedule conducts the resampling step every K timestamps for some $K > 1$. An adaptive schedule checks the effective sample size defined in (2.11) at each time and conducts resampling only when the effective sample size drops below a threshold, for example $0.3N$.

Without other specification, in the subsequent content of this thesis, the sequential

Algorithm 2: Sequential Importance Sampling with Resampling

Draw $\{x_0^{(i)}\}_{i=1}^N$ from $q_0(x_0)$;
Set $w_0^{(i)} \leftarrow 1$ for $i = 1, \ldots, N$;
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, N$ **do**
    Draw $x_t^{(i)}$ from $q_t(x_t \mid x_{0:t-1}^{(i)})$;
    Set

$$w_t^{(i)} \leftarrow w_{t-1}^{(i)} \frac{f_t(x_t^{(i)} \mid x_{0:t-1}^{(i)}) g_t(y_t \mid x_t^{(i)})}{q_t(x_t^{(i)} \mid x_{0:t-1}^{(i)})}; \tag{2.15}$$

  end
  if *Resampling* **then**
    Draw $j_1, \ldots, j_N$ from $[N]$ with probability $P[j_i = k] = \beta_k$;
    Set

$$x_{0:t}^{(i)} \leftarrow x_{0:t}^{(j_i)}, \quad w_t^{(i)} \leftarrow w_t^{(j_i)} / \beta_{j_i}.$$

  end
end
return $\{(x_{0:T}^{(i)}, w_T^{(i)})\}_{i=1}^N$.

Monte Carlo algorithm refers to the sequential importance sampling with resampling (SISR) shown in Algorithm 2.

2.2.4 Existing SMC Algorithms

Algorithm 2 provides the general framework of sequential Monte Carlo as the user has the flexibility in choosing arbitrary proposal distributions q_t used in importance sampling and arbitrary priority scores β used in resampling. In this section, I will list and discuss several well-known existing SMC algorithms and their choices of proposal distributions and priority scores.

The bootstrap particle filter (or Bayesian particle filter) (Kitagawa, 1996) uses the state equation as the proposal distribution such that $q_t(x_t \mid x_{0:t-1}) = f_t(x_t \mid x_{0:t-1})$. The weight increment is therefore $g_t(y_t \mid x_t)$. It comes from the Bayesian interpretation of the system by viewing the state equations as "prior" and the observation equation as "likelihood". This setting of proposal distributions works well when the observations are noisy, or equivalently, when the state equation is less volatile.

The independent particle filter (Lin et al., 2005) sets the observation equation as the

proposal distribution such that $q_t(x_t \mid x_{0:t-1}) \propto g_t(y_t \mid x_t)$. The correspondingly weight increment is proportional to $f_t(x_t \mid x_{0:t-1})$. This choice of proposal distribution is suitable for the cases when the observations are accurate compared with the state equations.

As a compromise between the bootstrap particle filter and the independent particle filter, Kong et al. (1994); Liu and Chen (1998) proposed to adopt $q_t(x_t \mid x_{0:t-1}) \propto f_t(x_t \mid x_{0:t-1})g_t(y_t \mid x_t)$ to reduce the χ^2-divergence between sampling distribution and the target distribution.

The auxiliary particle filter (Pitt and Shephard, 1999) suggests first conduct resampling with priority scores $\beta_t \propto w_t p(y_{t:t+\Delta} \mid x_{0:t-1})$ for a certain number of lookahead steps $\Delta > 0$ at time t, then drawing samples of x_t with proposal distribution $q(x_t \mid x_{0:t-1}) = p(x_t \mid x_{0:t-1}, y_{t:t+\Delta})$.

Extensions to the auxiliary particle filter are investigated as incorporating future observations improves the quality of SMC samples. For example, the twisted particle filter proposed by Whiteley and Lee (2014), the block sampling method in Doucet et al. (2006) and the family of look-ahead strategies reviewed in Lin et al. (2013). Specifically, the constrained SMC discussed in Chapter 3 can be viewed as a special case of the auxiliary particle filter as well, but with an estimated optimal priority score.

CHAPTER 3

Constrained Sequential Monte Carlo

3.1 Constrained Problems

3.1.1 Introduction

Stochastic dynamic systems often come with external observable information, including direct/indirect measurements, constraints and others. For example, in many physics and financial applications, one is often interested in the distribution of all possible paths of a given diffusion process with fixed starting and ending points (known as diffusion bridge) (Pedersen, 1995; Durham and Gallant, 2002; Lin et al., 2010). In protein structure studies, the properties of self-loops are often studied, where a self-loop is defined as a strand of proteins that forms a spatial loop by a chemical bond. Samples from the distribution of self-avoiding walks on a 3-D lattice with the same starting and ending points are often studied as a proxy of protein loops (Zhang et al., 2009; Lin et al., 2008a). The observations $y_{1:T}$ in the state space model discussed in Chapter 2 is another example.

In this section, we reformulate the constraints as an extension to the state space model defined by (2.2) and (2.3) such that more general constraints are considered. For example, some observations y_t can be missing from the state space model. The constraints can be any event in the corresponding sigma field of x_t such as $x_T > c$ for some constant c. In addition, the singular observation equation is allowed. For instance, $g_T(y_T \mid x_T) = \delta(y_T - x_T)$ with observed y_T corresponds to the fixed end-point constraint on x_T.

3.1.2 Stochastic Dynamic System with Constraints

Similar to the state space model, we assume $x_{0:T}$ is a sequence of unobserved random states, whose dynamics are governed by the state equations in (2.2). Instead of assuming a point observation y_t for each time t, we introduce the following general notation for external information/constraints. Let $\mathcal{I}_t$ be a piece of new information at time t, and $\mathcal{F}_t$ be the cumulative information imposed on the latent states up to time t. Hence, $\mathcal{F}_0 \supset \mathcal{F}_1 \supset \cdots \supset \mathcal{F}_T$ forms a sequence of monotonically non-increasing events, where $\mathcal{F}_t = \mathcal{F}_{t-1} \cap \mathcal{I}_t$. When there is no additional information at time t, we have $\mathcal{F}_t = \mathcal{F}_{t-1}$.

The posterior distribution (2.7) from the state space model is reformulated to

$$p(x_{0:T} \mid \mathcal{F}_T) \propto p(x_0, \mathcal{F}_0) \prod_{t=1}^{T} p(x_t, \mathcal{F}_t \mid x_{0:t-1}, \mathcal{F}_{t-1}), \tag{3.1}$$

where $p(x_t, \mathcal{F}_t \mid x_{0:t-1}, \mathcal{F}_{t-1}), t = 1, \ldots, T$, are specified by the system.

In the constrained sampling problems, some "strong" constraints are of particular interest. We define the strength of the cumulative constraints between t and $t+d$ with the χ^2-divergence measure

$$G(t, t+d) = D_{\chi^2}\Big(p(x_{0:t+d} \mid \mathcal{F}_{t+d}) \parallel p(x_{0:t+d} \mid \mathcal{F}_{t-1})\Big) := \int \frac{p^2(x_{0:t+d} \mid \mathcal{F}_{t+d})}{p(x_{0:t} \mid \mathcal{F}_{t-1})} dx_{0:t+d} - 1 \tag{3.2}$$

for $t > 0$ and $d \geqslant 0$. A large value of $G(t, t+d)$ reflects a strong impact of the constraints imposed between t and $t+d$ (inclusive).

3.1.3 Special Cases

Several special cases of the constrained system will be discussed in this section.

Case 1: Frequent and weak constraints: The traditional state space model with a noisy observation y_t for each time t belongs to this special constrained system. Suppose the observation at time t has the following random mapping representation $y_t = \tilde{g}_t(x_t, \nu_t)$, where ν_t is a random variable that used to match the conditional distribution $g_t(y_t \mid x_t)$. The constraint is in the form $\mathcal{I}_t = \{y_t = \tilde{g}_t(x_t, \nu_t)\}$ and the event $\mathcal{F}_t = \{\tilde{g}_s(x_s, \nu_s) = y_s, s = 0, \ldots, t\}$

for $t = 0, \cdots, T$. In this case,

$$p(x_t, \mathcal{F}_t \mid x_{0:t-1}, \mathcal{F}_{t-1}) = f_t(x_t \mid x_{t-1}) g_t(y_t \mid x_t),$$

where f_t and g_t are the state equation and observation equation from (2.2) and (2.3) correspondingly. The frequent observations continuously provide information about the underlying process $x_{0:T}$. The conventional sequential Monte Carlo depicted in Algorithm 2 has been extensively studied to solve this frequent and weak constraints case.

Case 2: Rare and strong constraints: The diffusion bridge problems (Lin et al., 2010) fall into this case, where the problem is to generate paths that connect two fixed endpoints $x_0 = a$ and $x_0 = b$. The corresponding constraint events are $\mathcal{F}_0 = \cdots = \mathcal{F}_{T-1} = \{x_0 = a\}$ and $\mathcal{F}_T = \{x_0 = a, x_T = b\}$. The constraint at time T, $\mathcal{I}_T = \{x_T = b\}$, is extremely strong as the strength are

$$G(1, T) = \cdots = G(T-1, T) = \infty.$$

Case 3: Periodic and intermediate constraints: When there are noisy measurements of the latent states $x_{0:T}$ periodically, we assume that $T = KM$, and y_k is a noisy measurement of x_{kM} for $k = 1, \cdots, K$. The intermediate observations split the whole path into K segments as shown in Figure 3.1. This system can be viewed as a sequence of connected diffusion bridge problems. Specifically, The segment $(x_0, \cdots, x_M, y_t)$ is the first diffusion bridge problem, where the state equations are known and the endpoints x_0 and y_1 are fixed. Any subsequent segments are imposed with fixed endpoint constraints as well, since y_k is viewed as the fixed point, $p(y_k \mid x_{kM})$ is treated as the state equation of the last step and $p(x_{(k-1)M} \mid y_{1:k-1})$ obtained from previous segment is the initial distribution of $x_{(k-1)M}$. The samples of this periodic strong constraints can be drawn in a segment-wise fashion.

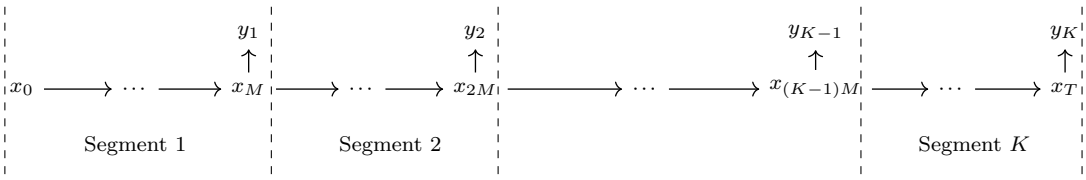


Figure 3.1: Segmentation of a stochastic process with intermediate observations.

Case 4: Multilevel constraints: In some applications, there may exist multiple levels

of constraints, including those with a hierarchical structure, such as one level of weak but frequent constraints and another level of strong but rare constraints (a hybrid setting of Case 1 and Case 2). A special case is a standard state space model with two fixed endpoint constraints $x_0 = a$ and $x_T = b$. The target posterior distribution is now $p(x_{0:T} \mid x_0 = a, y_{1:T-1}, x_T = b)$. The routine observations $y_1, \cdots, y_{T-1}$ can be viewed as a layer of weak constraints and the fixed point constraints are viewed as a layer of strong constraints.

3.2 Existing SMC-based Approaches

In this section, several existing SMC algorithms designed for the diffusion bridge problem are reviewed. The diffusion bridge problem (Durham and Gallant, 2002; Lin et al., 2010) imposes two fixed endpoint constraints $x_0 = a$ and $x_T = b$ to an underlying diffusion process. The constraint is strong when the two distributions, $p(x_{1:T-1} \mid x_0 = a)$ and $p(x_{1:T-1} \mid x_0 = a, x_T = b)$ have a large divergence, or equivalently, a large value of $G(T-1, T)$.

Pedersen (1995) proposed to generate the samples through the underlying diffusion process without considering the endpoint constraint and then force the samples to connect with the fixed endpoint at the end. It works when $G(T-1, T)$ is small as in the last step, the forced connection does not reduce effective sample size significantly. However, when the constraint is strong, it may not be efficient due to the large deviation of the end of the forward paths from the enforced end point.

To avoid the potential large deviation at the endpoint, Durham and Gallant (2002) proposed to employ importance sampling to guide the sample paths to the fixed endpoint proactively. Specifically, the proposal distribution at each time t is a modified version of the underlying diffusion process by adding a drift term, $(b - x_{t-1}^{(i)})\Delta t/(T - t + 1)$, where $x_{t-1}^{(i)}$ is the current value of state and Δt is the time step in the discretization. The proposal distribution with a linear interpolation drift term turns the joint proposal distribution $\prod_{t=1}^{T-1} q_t(x_t \mid x_{t-1})$ to a linear diffusion process, where the drift term is a linear function of the current state x_t. Although the sample paths generated using linear diffusion proposal distributions are weighted properly using SMC, it breaks the underlying dynamics when the state equations $f_t(x_t \mid x_{0:t-1})$ are nonlinear. In other words, the sample paths may not

be representative and efficient. See the discussion of Lin et al. (2010).

Lin et al. (2010) proposed a resampling strategy that keeps the underlying dynamics of the latent process and at the same time adjust the samples proactively according to the fixed endpoint constraint. Specifically, Lin et al. (2010) suggests to use the dynamics $f_t(x_t \mid x_{t-1})$ as the proposal distribution, preserving any potential nonlinear properties. In the resampling step, the priority score is designed to incorporate future information — the endpoint constraint. The priority score proposed by Lin et al. (2010) for the diffusion bridge problems is $\beta_t \propto w_t p(x_T = b \mid x_t)$, which measures the probability of current state reaching the fixed endpoint. A backward pilot procedure is conducted as a pre-processing step to estimate the priority scores. The backward pilot resampling strategy achieves good efficiency. It improves the forward sampling by bringing future information and constraints with minimum additional computational costs.

3.3 Constrained Sequential Monte Carlo

3.3.1 Distributions in SMC

In this section, we discuss several important distributions used in SMC and re-state the SMC concepts in Chapter 2 with the distributions.

Let $\mathbb{Q}_0(x_0), \mathbb{Q}_1(x_{0:1}), \cdots, \mathbb{Q}_T(x_{0:T})$ be the sequence of forward propagation distributions such that at each time t, the SMC samples $\{(x_{0:t}^{(i)}, w_t^{(i)})\}_{i=1}^N$ are properly weighted with respect to $\mathbb{Q}_t(x_{0:t})$. For the SMC algorithm in Algorithm 2, $\mathbb{Q}_t(x_{0:t}) = f_0(x_0)\prod_{t=1}^T f_t(x_t \mid x_{0:t-1})$. The weight update step (2.15) can be rewritten to

$$w_t^{(i)} \leftarrow w_{t-1}^{(i)} \frac{\mathbb{Q}_t(x_{0:t}^{(i)})}{\mathbb{Q}_{t-1}(x_{0:t-1}^{(i)}) q_t(x_t^{(i)} \mid x_{0:t-1}^{(i)})}. \tag{3.3}$$

The formula (3.3) is more general in the sense that the SISR algorithm is still valid as long as the sequence of distributions $\mathbb{Q}_t$ are defined even when y_t is not observed.

Conventional SMC approaches (Gordon et al., 1993; Liu and Chen, 1998) set the forward

propagation distributions $\mathbb{Q}_t(x_{0:t})$ to

$$\mathbb{P}_t^0(x_{0:t}) = p(x_{0:t} \mid \mathcal{F}_t), \quad t = 0, 1, \ldots, T, \tag{3.4}$$

using all information up to time t. As discussed in Section 3.2, using $\mathbb{P}_t^0$ as the forward propagation works well for frequent and weak constraints, but is not efficient when strong constraints exist (for example, the diffusion bridge case).

Since $\mathbb{P}_t^0$ ignores all future information, it is natural to consider another extreme — the forward propagation distribution that incorporates ALL future information. Specifically, we define such a distribution by

$$\mathbb{P}_t(x_{0:t}) = p(x_{0:t} \mid \mathcal{F}_T) = p(x_0 \mid \mathcal{F}_T) \prod_{s=1}^{t} p(x_s \mid x_{0:s-1}, \mathcal{F}_T), \tag{3.5}$$

for $t = 0, 1, \ldots, T$. It is the most efficient algorithm if one was able to sample from the conditional posterior distribution $p(x_s \mid x_{0:s-1}, \mathcal{F}_T)$ exactly. However, in most cases, $p(x_s \mid x_{0:s-1}, \mathcal{F}_T)$ is infeasible and is difficult to sample from as it involves a high-dimensional integral

$$p(x_s \mid x_{0:s-1}, \mathcal{F}_T) \propto \int \cdots \int \prod_{s=t}^{T} p(x_s, \mathcal{F}_s \mid x_{0:s-1}, \mathcal{F}_{s-1}) dx_{t+1} \cdots dx_T. \tag{3.6}$$

As a compromise between the optimality of $\mathbb{P}_t$ in (3.5) and the feasibility of $\mathbb{P}_t^0$ in (3.4), we propose to adopt the following forward propagation measure

$$\mathbb{P}_t^*(x_{0:t}) = p(x_{0:t} \mid \mathcal{F}_{t_+}), \quad t = 0, 1, \ldots, T, \tag{3.7}$$

where $t_+ \geqslant t$ is the next time when a strong constraint is imposed after time t (inclusive), or the cumulative strength of the constraint $G(t, t_+)$ defined in (3.2) exceeds certain threshold. In practice, the selection of t_+ could depend on specific problems and be user-defined. For instance, in the diffusion bridge sampling problem, we use $t_+ = T$. In the conventional state space model with frequent observations at every time, we may use $t_+ = \min\{t + d, T\}$ for some constant delay d as in the delayed SMC algorithm (Lin et al., 2013). In the periodic observation case (Case 3 in Section 3.1.3), t_+ is where we insert y_k in the process.

By observing the fact that in most cases the next available strong constraint plays an important role in shaping the path distribution, the proposed forward propagation distribution $\mathbb{P}_t^*$ works as a proxy of $\mathbb{P}_t$ but is more feasible as we will discuss later.

3.3.2 Forward Propagation using $\mathbb{P}_t^*$

As discussed in the previous section, the compromised distribution $\mathbb{P}_t^*(x_{0:t})$ balances the use of future constraints and the computational efficiency. To use $\mathbb{P}_t^*(x_{1:t})$ as the propagation distribution $\mathbb{Q}_t(x_{1:T})$ in SMC, it is ideal to draw samples from the exact distribution such that $q_t(x_t \mid x_{0:t-1}) = \mathbb{P}_t^*(x_t \mid x_{0:t-1}) = p(x_t \mid x_{0:t-1}, \mathcal{F}_{t_+})$. However, it is usually difficult, especially when t_+ is far away from t, since it involves a high dimensional integral similar to (3.6). On the other hand, $\mathbb{P}_t^0$ is often easier to work with, with proposal distributions equal or close to $\mathbb{P}_t^0(x_t \mid x_{0:t=1}) = p(x_t \mid x_{0:t-1}, \mathcal{F}_t)$. Notice that

$$\mathbb{P}_t^*(x_{0:t}) \propto \mathbb{P}_t^0(x_{0:t}) p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t),$$

where $(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t)$ is the Radon-Nikodym derivative between $\mathbb{P}_t^*$ and $\mathbb{P}_t^0$ up to a constant. Therefore, a properly weighted sample set under the distribution $\mathbb{P}_t^0$ can be easily changed to sample set properly weighted with respect to $\mathbb{P}_t^*$ by multiplying the weights obtained under $\mathbb{P}_t^0$ by a factor $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t)$.

Based on the above observations, instead of propagate SMC samples with respect to the infeasible distribution $p(x_t \mid x_{0:t-1}, \mathcal{F}_{t_+})$, we proposed to propagate with respect to $\mathbb{P}_t^0$ and then resample with priority score

$$\beta_t^{(i)} = w_t^{(i)} p(\mathcal{F}_{t_+} \mid x_{0:t}^{(i)}, \mathcal{F}_t) \tag{3.8}$$

to adjust the distribution of samples. $\beta_t^{(i)}$ in (3.8) is called the *optimal priority score* because the samples will approximately follow $\mathbb{P}_t^*$ after resampling. A heatmap of the term $p(\mathcal{F}_{t_+} \mid x_{0:t}^{(i)}, \mathcal{F}_t)$ in a Markovian nonlinear process as a function of time t and the state value x_t is plotted in Figure 3.2.

Not only does incorporating $\mathbb{P}_t^*$ in the priority score make the propagation step possible

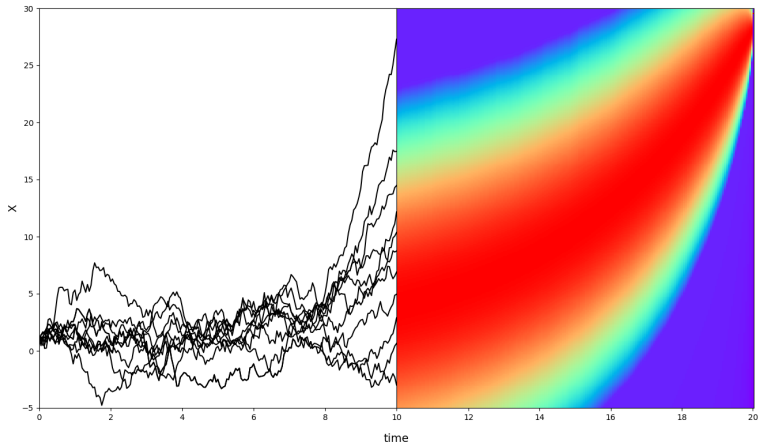


Figure 3.2: Illustration of the resampling step at time $t = 10$ in SMCc. The left side shows several forward paths $x_{0:t}^{(i)}$ to be resampled, and the right side shows the heatmap of $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t)$.

using $\mathbb{P}_t^0$, but also avoids computation of the exact value of the probability $p(\mathcal{F}_{t_+} \mid x_{0:t}^{(i)}, \mathcal{F}_t)$ in the weight update step. As mentioned in Chapter 2, the choice of priority score is arbitrary and the weighted samples are still properly weighted even when the optimal priority score (3.8) is replaced with an approximated value. We refer to this method as the sequential Monte Carlo with constraints (SMCc) method. The details of the algorithm are depicted in Algorithm 3, where $\hat{p}(\mathcal{F}_{t_+} \mid x_{0:t}^{(i)}, \mathcal{F}_t)$ is the approximated value of $p(\mathcal{F}_{t_+} \mid x_{0:t}^{(i)}, \mathcal{F}_t)$. We will discuss how to approximate it in the next section.

3.4 Estimation of the Optimal Priority Score

In this section, we omit the trivial case that $t_+ = t$, in which $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = 1$, and focus on the case that $t_+ > t$. Then the second term in the optimal priority score (3.8) can be written as

$$p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = \int \cdots \int \prod_{s=t+1}^{t_+} p(x_s, \mathcal{F}_s \mid x_{0:s-1}, \mathcal{F}_{s-1}) dx_{t+1} \cdots dx_{t_+}, \tag{3.9}$$

which often does not have a closed-form solution. In this section, three methods to approximate $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t)$ are presented: the parametric approach (SMCc-PA), the forward pilot

Algorithm 3: Sequential Monte Carlo with Constraints (SMCc)

Draw $\{x_0^{(i)}\}_{i=1}^N$ from $q_0(x_0)$;
Set $w_0^{(i)} \leftarrow 1$ for $i = 1, \ldots, N$;
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, N$ **do**
    Draw $x_t^{(i)}$ from $q_t(x_t \mid x_{0:t-1}^{(i)})$;
    Set

$$w_t^{(i)} \leftarrow w_{t-1}^{(i)} \frac{p(x_t^{(i)}, \mathcal{F}_t \mid x_{0:t-1}^{(i)}, \mathcal{F}_{t-1})}{q_t(x_t^{(i)} \mid x_{0:t-1}^{(i)})};$$

  end
  (Optional) Resampling with priority score

$$\beta_t^{(i)} \propto w_t^{(i)} \hat{p}(\mathcal{F}_{t_+} \mid x_{0:t}^{(i)}, \mathcal{F}_t).$$

end
return $\{(x_{0:T}^{(i)}, w_T^{(i)})\}_{i=1}^N$.

approach (SMCc-FP) and the backward pilot approach (SMCc-BP).

Note that we focus on the cases with rare and strong constraints. We assume a pre-fixed time stamps of the constraints: $0 = T_0 < T_1 < \cdots < T_K = T$. Then $t_+ = T_k$ if $T_{k-1} < t \leqslant T_k$.

3.4.1 Parametric Approximation

One may assume a parametric form for $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t)$ based on some prior knowledge. For example, Zhang et al. (2007); Lin et al. (2008b) used SMCc approach with $t_+ = T$ to generate protein conformation samples with certian distance constraints, where $p(\mathcal{F}_T \mid x_{0:t}, \mathcal{F}_t)$ is approximated with parametric functions based on the distance between x_t and x_T.

Scharth and Kohn (2016) proposed the particle efficient importance sampling (PEIS) method, which approximates $p(x_t \mid x_{0:t-1}, \mathcal{F}_T)$ and the optimal priority score β_t within parametric families though an iterative local optimization routine. For simplicity, assume the system is Markovian such that $p(x_t, \mathcal{F}_t \mid x_{0:t-1}, \mathcal{F}_{t-1}) = p(x_t, \mathcal{F}_t \mid x_{t-1}, \mathcal{F}_{t-1})$ for all t. PEIS assumes

$$q(x_t \mid x_{t-1}; \theta_t) = \psi_t(x_t, x_{t-1}; \theta_t) / \chi_t(x_{t-1}; \theta_t),$$

where $\psi_t(x_t, x_{t-1}; \theta_t)$ is in a parametric family with parameter θ_t, and

$$\chi_t(x_{t-1}; \theta_t) = \int \psi_t(x_t, x_{t-1}; \theta_t) dx_t$$

is the normalizing term. With these notations, the importance weight at time T becomes

$$\begin{aligned} w_T(x_{0:T}) &= \frac{p(x_{0:T} \,|\, \mathcal{F}_T)}{q(x_0; \theta_0) \prod_{t=1}^T q(x_t \,|\, x_{t-1}; \theta_t)} \\ &\propto \frac{p(x_0, \mathcal{F}_0) \prod_{t=1}^T p(x_t, \mathcal{F}_t \,|\, x_{t-1}, \mathcal{F}_{t-1})}{q(x_0; \theta_0) \prod_{t=1}^T q(x_t \,|\, x_{t-1}; \theta_t)} \\ &\propto \frac{p(x_0, \mathcal{F}_0) \chi_1(x_0; \theta_1)}{\psi_0(x_0; \theta_0)} \left[\prod_{t=1}^{T-1} \frac{p(x_t, \mathcal{F}_t \,|\, x_{t-1}, \mathcal{F}_{t-1}) \chi_{t+1}(x_t; \theta_{t+1})}{\psi_t(x_t, x_{t-1}; \theta_t)} \right] \frac{p(x_T, \mathcal{F}_T \,|\, x_{T-1}, \mathcal{F}_{T-1})}{\psi_T(x_T, x_{T-1}; \theta_T)}, \end{aligned} \tag{3.10}$$

where the initial $\phi_0(x_0; \theta_0)$ is also restricted to a parametric family. Note that w_T is linked to the effective sample size defined in (2.11). Minimizing the variation of w_T in (3.10) can be done through a backward iterative optimization procedure.

We start with an optimal θ_T that minimizes the variation of the term

$$\frac{p(x_T, \mathcal{F}_T \,|\, x_{T-1}, \mathcal{F}_{T-1})}{\psi_T(x_T, x_{T-1}; \theta_T)}.$$

Then going backward recursively for $t = T-1, \ldots, 1$, we find θ_t that minimizes the variation of

$$\frac{p(x_t, \mathcal{F}_t \,|\, x_{t-1}, \mathcal{F}_{t-1}) \chi_{t+1}(x_t; \theta_{t+1})}{\psi_t(x_t, x_{t-1}; \theta_t)}.$$

The PEIS method can be easily adapted to our setting to approximate $p(\mathcal{F}_{t_+} \mid x_t, \mathcal{F}_t)$ as shown in Algorithm 4. The optimization procedure is repeated for each time interval from $T_{k-1}+1$ to T_k. The normalizing term $\chi_{t+1}(x_t; \theta_{t+1})$ is used as an approximation of $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = p(\mathcal{F}_{t_+} \mid x_t, \mathcal{F}_t)$. The performance of this method greatly depends on the chioce of the parametric family.

3.4.2 Approximation Based on Forward Pilots

When there does not exist appropriate parametric families that can approximate $p(x_t, \mathcal{F}_t \mid x_{0:t-1}, \mathcal{F}_{t-1})$, non-parametric and Monte Carlo approaches should be considered instead.

Algorithm 4: PEIS Parameter Optimization

for $k = 1, \ldots, K$ **do**

  Initialize the parameters $\theta_t^{[0]}$ for $t = T_{k-1}+1, \cdots, T_k$;

  repeat

    Generate samples $x_{T_{k-1}:T_k}^{(j)}$, $j = 1, \cdots, m$, from the proposal distribution

$$q(x_{T_{k-1}}) \prod_{t=T_{k-1}+1}^{T_k} \frac{\psi_t(x_t, x_{t-1}; \theta_t^{[l-1]})}{\chi_t(x_{t-1}; \theta_t^{[l-1]})},$$

    where $q(x_{T_{k-1}})$ is a distribution close to $p(x_{T_{k-1}} \,|\, \mathcal{F}_{T_{k-1}})$;

    Calculate the weights

$$w_{T_k}^{(j)} = \frac{p(\mathcal{I}_{T_{k-1}} \,|\, x_{T_{k-1}}^{(j)})}{q(x_{T_{k-1}}^{(j)})} \prod_{t=T_{k-1}+1}^{T_k} \frac{p(x_t^{(j)} \,|\, x_{t-1}^{(j)}) p(\mathcal{I}_t \,|\, x_t^{(j)})}{q(x_t^{(j)} \,|\, x_{t-1}^{(j)}; \theta_t^{[l-1]})}.$$

    for $t = T_k, T_k - 1, \cdots, T_{k-1}+1$ **do**

      solve the minimization problem

$$(\theta_t^{[l]}, \gamma_t^{[l]}) = \arg\min_{\theta, \gamma} \sum_{j=1}^{m} w_{T_k}^{(j)} \Big\{ \log \left[p(x_t^{(j)} \,|\, x_{t-1}^{(j)}) p(\mathcal{I}_t \,|\, x_t^{(j)}) \chi_{t+1}(x_t^{(j)}; \theta_{t+1}^{[l]}) \right] - \gamma - \log \left[\psi_t(x_t^{(j)}, x_{t-1}^{(j)}; \theta) \right] \Big\}^2,$$

      where $\chi_{t+1}(x_t; \theta_{t+1})$ is set to a constant when $t = T_K$;

    end

  until *convergence*;

end

Let θ_t^*, $t = T_{k-1}+1, \cdots, T_k$, be the converged parameters;

return the estimated functions

$$\left\{ \widehat{p}(\mathcal{F}_{t_+} \,|\, x_{0:t}, \mathcal{F}_t) = \chi_{t+1}(x_t; \theta_{t+1}^*) \right\}_{t = T_{k-1}+1, \cdots, T_k - 1; k = 1, \cdots, K}$$

One of the approaches is to send out forward pilot samples, which has been proposed by Wang et al. (2002); Zhang and Liu (2002) and is used for delayed estimation in Lin et al. (2013).

Suppose that at time t we have samples $\{(x_{0:t}^{(i)}, w_t^{(i)})\}_{i=1}^N$ properly weighted with respect to $\mathbb{P}_t^0$. In the traditional way of forward pilot sampling, the pilot samples $\tilde{x}_{t+1:t_+}^{(i,j)} = (\tilde{x}_{t+1}^{(i,j)}, \cdots, \tilde{x}_{t_+}^{(i,j)})$, $j = 1, \cdots, J$ are generated for each sample $x_{0:t}^{(i)}$ from a (joint) proposal dis-

tribution $\prod_{s=t+1}^{t_+} \varphi(x_s \mid x_{0:t}^{(i)}, x_{t+1:s-1})$ and are weighted by $U_t^{(i,j)} = \prod_{s=t+1}^{t_+} u_s^{(i,j)}$ with

$$u_s^{(i,j)} = \frac{p(\widetilde{x}_s^{(i,j)}, \mathcal{F}_s \mid x_{0:t}^{(i)}, \widetilde{x}_{t+1:s-1}^{(i,j)})}{\varphi(\widetilde{x}_s^{(i,j)} \mid x_{0:t}^{(i)}, \widetilde{x}_{t+1:s-1}^{(i,j)})}.$$

By observing $\mathbb{E}[U_t^{(i,j)} \mid x_{0:t}^{(i)}] = p(\mathcal{F}_{t_+} \mid x_{0:t}^{(i)}, \mathcal{F}_t)$, the latter can therefore be approximated by $J^{-1} \sum_{j=1}^{J} U_t^{(i,j)}$. The computational burden of the traditional forward pilot methods is high as it requires the generation of pilot samples for every path $x_{0:t}^{(i)}$ at every time t.

Here we propose a new forward pilot approach to significantly reduce the computational cost using nonparametric smoothing technique. In addition, we suppose there exists a low dimensional statistics $S(x_{0:t})$ that summarizes $x_{0:t}$ in the way that

$$p(x_{t:t+d}, \mathcal{F}_{t+d} \mid x_{0:t-1}, \mathcal{F}_{t-1}) = p(x_{t:t+d}, \mathcal{F}_t \mid S(x_{0:t-1}), \mathcal{F}_{t-1})$$

for all t and $d = 0, 1, \ldots,$ and suppose there exists a function $\phi(\cdot)$ such that $S(x_{0:t}) = \phi(S(x_{0:t-1}), x_t)$. Then we can work on a low dimensional space such that $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = p(\mathcal{F}_{t_+} \mid S(x_{0:t}), \mathcal{F}_t)$ is a function of $S(x_{0:t})$. The algorithm is presented in Algorithm 5.

Note that for $U_t^{(j)} = \prod_{s=t+1}^{T_k} \tilde{u}_s^{(j)}$ defined in Algorithm 5, we have

$$\mathbb{E}[U_t^{(j)} \mid S_t^{(j)} = S] = p(\mathcal{F}_{t_+} \mid S(x_{0:t}) = S, \mathcal{F}_t)$$

for all t. Therefore, $p(\mathcal{F}_{t_+} \mid S(x_{0:t}, \mathcal{F}_t)$ can be estimated by $\{(U_t^{(j)}, S_t^{(j)})\}_{j=1}^{M}$ nonparametrically. Algorithm 5 choose to use the histogram method instead of kernel smoothing method in order to control the computational cost. Compared with the traditional pilot sampling method in Wang et al. (2002), Algorithm 5 only need to be conducted once to obtain $\hat{p}(\mathcal{F}_+ \mid S(x_{0:t}), \mathcal{F}_t)$ for all t.

The proposal distribution $\varphi(\cdot)$ is crucial to the accuracy of Algorithm 5 because the forward pilot samples need to comply with the constraint $\mathcal{I}_{T_k}$ finally. It is suggested to incorporate $\mathcal{I}_{T_k}$ into $\varphi(\cdot)$ such that the pilot samples have a reasonable large probability to satisfy the constraint $\mathcal{I}_{T_k}$.

Algorithm 5: Forward Pilot Smoothing

for $k = 1, \ldots, K$ **do**
  for $j = 1, \ldots, M$ **do**
    Draw samples $S_{T_{k-1}}^{(j)}$ from a proposal distribution $\varphi(S)$ that covers the support of $S(x_{0:T_{k-1}})$;
  end
  for $t = T_{k-1}+1, \cdots, T_k$ **do**
    for $j = 1, \ldots, M$ **do**
      Generate samples $\widetilde{x}_t^{(j)}$ from a proposal distribution $\varphi(\widetilde{x}_t \mid S_{t-1}^{(j)})$;
      Calculate $S_t^{(j)} = \phi(S_{t-1}^{(j)}, \widetilde{x}_t^{(j)})$;
      Calculate the incremental weights

$$\widetilde{u}_t^{(j)} = \frac{p(\widetilde{x}_t^{(j)}, \mathcal{F}_t \mid S(\widetilde{x}_{0:t-1}^{(j)}) = S_{t-1}^{(j)}, \mathcal{F}_{t-1})}{\varphi(\widetilde{x}_t^{(j)} \mid S_{t-1}^{(j)})},$$

    end
  end
  for $t = T_{k-1}+1, \cdots, T_k$ **do**
    for $j = 1, \ldots, M$ **do**
      Compute $U_t^{(j)} = \prod_{s=t+1}^{T_k} \widetilde{u}_s^{(j)}$;
    end
    Let $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_D$ be a partition of the support of $S(x_{0:t})$;
    Estimate $p(\mathcal{F}_{t_+} \,|\, x_{0:t}, \mathcal{F}_t) = p(\mathcal{F}_{t_+} | S(x_{0:t}), \mathcal{F}_t)$ by

$$h_t(S(x_{0:t})) = \sum_{d=1}^{D} \xi_{t,d} \mathbb{I}(S(x_{0:t}) \in \mathcal{S}_d)$$

    with

$$\xi_{t,k} = \frac{\sum_{j=1}^{m} U_t^{(j)} \mathbb{I}(S_t^{(j)} \in \mathcal{S}_d)}{\sum_{j=1}^{m} \mathbb{I}(S_t^{(j)} \in \mathcal{S}_d)},$$

    where $\mathbb{I}(\cdot)$ is the indicator function;
  end
end
return the estimated functions

$$\left\{ \widehat{p}(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = h_t(S(x_{0:t})) \right\}_{t=T_{k-1}+1, \cdots, T_k - 1; k=1, \cdots, K}$$

3.4.3 Approximation Based on Backward Pilots

As discussed in the previous section, the new forward pilot approach still requires a careful design of the proposal distribution used in sampling the pilot especially when the constraint is strong. It is nature to consider the opposite direction — sampling backwards from the time T_k so that the constraint $\mathcal{I}_{T_K}$ is enforced at the very beginning. In order to sampling pilots in a backward direction, we often require the sampling of x_{T_k-1} conditioned on x_{T_k} does not depend on the earlier states $x_{0:T_k-2}$. Therefore, we assume the underlying dynamic system is Markovian such that

$$p(x_t, \mathcal{I}_t \mid x_{0:t-1}, \mathcal{F}_{t-1}) = p(x_t, \mathcal{I}_t \mid x_{t-1})$$

for all t. Consequently, $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = p(\mathcal{I}_{t+1:t_+} \mid x_t)$ is a function of x_t and does not depend on the past information before time t. Here $\mathcal{I}_{t+1:t_+}$ denotes the cumulative constraints imposed between time $t+1$ and t_+. Here we generalize the backward pilot strategy in Lin et al. (2010) to the constrained problem of interest. The algorithm is presented in Algorithm 6.

The importance weight for the backward pilot $\tilde{x}_{t:t_+}$ in Algorithm 6 is

$$\widetilde{w}_t = \frac{p(\widetilde{x}_{t+1:t_+}, \mathcal{I}_{t+1:t_+} \mid \widetilde{x}_t)}{r(\widetilde{x}_{t:t_+})},$$

where $r(\tilde{x}_{t:t_+})$ is the joint proposal distribution to generate the backward pilots. By taking expectation conditioned on $\tilde{x}_t$, we have

$$\begin{aligned}\mathbb{E}(\widetilde{w}_t \mid \widetilde{x}_t) &= \int \cdots \int \frac{p(\widetilde{x}_{t+1:t_+}, \mathcal{I}_{t+1:t_+} \mid \widetilde{x}_t)}{r(\widetilde{x}_t, \widetilde{x}_{t+1:t_+})} r(\widetilde{x}_{t+1:t_+} \mid \widetilde{x}_t) d\widetilde{x}_{t+1:t_+} \\ &= p(\mathcal{I}_{t+1:t+} | \widetilde{x}_t) / r(\widetilde{x}_t),\end{aligned}$$

where $r(\widetilde{x}_{t+1:t_+} \mid \widetilde{x}_t)$ and $r(\widetilde{x}_t)$ are the conditional distribution and the marginal distribution induced from $r(\widetilde{x}_{t:t_+})$, respectively. Therefore,

$$p(\mathcal{I}_{t+1:t+} | \widetilde{x}_t) = r(\widetilde{x}_t) E(\widetilde{w}_t \mid \widetilde{x}_t),$$

Algorithm 6: Backward Pilot Smoothing

for $k = 1, \ldots, K$ **do**
  for $j = 1, \ldots, M$ **do**
    Draw samples $\tilde{x}_{T_k}^{(j)}$ from a proposal distribution $r(x_{T_k})$ approximately proportional to $p(\mathcal{I}_{T_k} \mid x_{T_k})$;
    Set $\tilde{w}_{T_k}^{(j)} \leftarrow 1/r(\tilde{x}_{T_k}^{(j)})$;
  end
  for $t = T_k - 1, \cdots, T_{k-1} + 1$ **do**
    for $j = 1, \ldots, M$ **do**
      Generate samples $\tilde{x}_t^{(j)}$ from a proposal distribution $r(\tilde{x}_t \mid \tilde{x}_{t+1}^{(j)})$;
      Calculate the incremental weights

$$\widetilde{u}_t^{(j)} = \frac{p(\tilde{x}_{t+1}^{(j)}, \mathcal{I}_{t+1} \mid \tilde{x}_t^{(j)})}{r(\tilde{x}_t^{(j)} \mid \tilde{x}_{t+1}^{(j)})},$$

    end
    Let $\mathcal{X}_1 \cup \cdots \cup \mathcal{X}_D$ be a partition of the support of x_t;
    Estimate $p(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = p(\mathcal{I}_{t+1:t_+} \mid x_t)$ by

$$h_t(x_t) = \sum_{d=1}^{D} \eta_{t,d} \mathbb{I}(x_t \in \mathcal{X}_d),$$

    where

$$\eta_{t,d} = \frac{1}{m|\mathcal{X}_d|} \sum_{j=1}^{m} \widetilde{w}_t^{(j)} \mathbb{I}(\widetilde{x}_t^{(j)} \in \mathcal{X}_d),$$

    and $|\mathcal{X}_d|$ denotes the volume of the subset $\mathcal{X}_d$;
  end
end
return the estimated functions $\left\{\widehat{p}(\mathcal{F}_{t_+} \mid x_{0:t}, \mathcal{F}_t) = h_t(x_t)\right\}_{t=T_{k-1}+1,\cdots,T_k-1;k=1,\cdots,K}$

which can be estimated by the nonparametric density estimator of $\{(\tilde{x}_t^{(j)}, \tilde{w}_t^{(j)})\}_{j=1}^M$ at any $\tilde{x}_t$.

Compared with the forward pilot method, the backward pilots are generated backwards, starting from the constrained time point T_k. Since the strong constraint $\mathcal{I}_{T_k}$ is automatically incorporated in the proposal distribution to generate $\tilde{x}_{T_k}$ at the beginning, it is often expected to have a more accurate approximation estimation of $p(\mathcal{I}_{t+1:t_+} \mid x_t)$. However, it requires the system to be Markovian.

3.5 Example: System with Intermediate Observations

In this example, we consider a diffusion process $\{X_\lambda\}_{0\le\lambda\le 90}$ governed by the following stochastic differential equation as discussed in (Beskos et al., 2006)

$$dX_\lambda = \sin(X_\lambda - \pi)d\lambda + dW_\lambda,$$

where W_λ is a standard Brownian motion. By inserting states x_t at the time points $\lambda_t = t\nu$, $t = 0, 1, \cdots, T$ with $T = 90/\nu$, the continuous-time diffusion process $\{X_\lambda\}_{0\le\lambda\le 90}$ can be approximated by the discrete-time process $\{x_0, x_1, \cdots, x_T\}$. We adopt the Euler-Maruyama approximation and the discretized version of the continuous-time diffusion process can be now written as

$$x_t = x_{t-1} + \nu \sin(x_{t-1} - \pi) + \varepsilon_t, \tag{3.11}$$

where $\varepsilon_t \sim N(0, \nu)$. We take $\nu = 0.1$ in this example.

In this simulation study, we assume two noisy observations of X_λ are made at times $\lambda = 30$ and $\lambda = 60$. That is,

$$Y_{30} \sim N(X_{30}, \sigma^2) \quad \text{and} \quad Y_{60} \sim N(X_{60}, \sigma^2),$$

where σ^2 is the noisy level which controls the accuracy of these two observations. In addition, we also fix the two endpoints at $X_0 = a$ and $X_{90} = b$. The discretized time points $T_0 = 0$, $T_1 = 30/\nu$, $T_2 = 60/\nu$ and $T_3 = 90/\nu$ are considered to have strong constraints. The

SMCc-BP method is applied to generate sample paths of $x_{0:T}$ conditional on the constraints $(X_0, Y_{30}, Y_{60}, X_{90})$. That is, we utilize Algorithm 3 to generate samples, and the backward pilot smoothing algorithm in Algorithm 6 is used to compute the approximated optimal resampling priority scores. The state dynamics equation (3.11) is used as the proposal distribution in generating forward paths. The backward pilots are generated according to Algorithm 6 with the proposal distribution $r(\widetilde{x}_t \mid \widetilde{x}_{t+1}) \sim N(\widetilde{x}_{t+1} - \nu \sin(\widetilde{x}_{t+1} - \pi), \nu)$. Resampling is conducted dynamically when ESS_t defined in (2.11) is less than $0.3n$. In this example, the time line is split into three segments. The segmental sampling procedure is demonstrated in Figure 3.3.

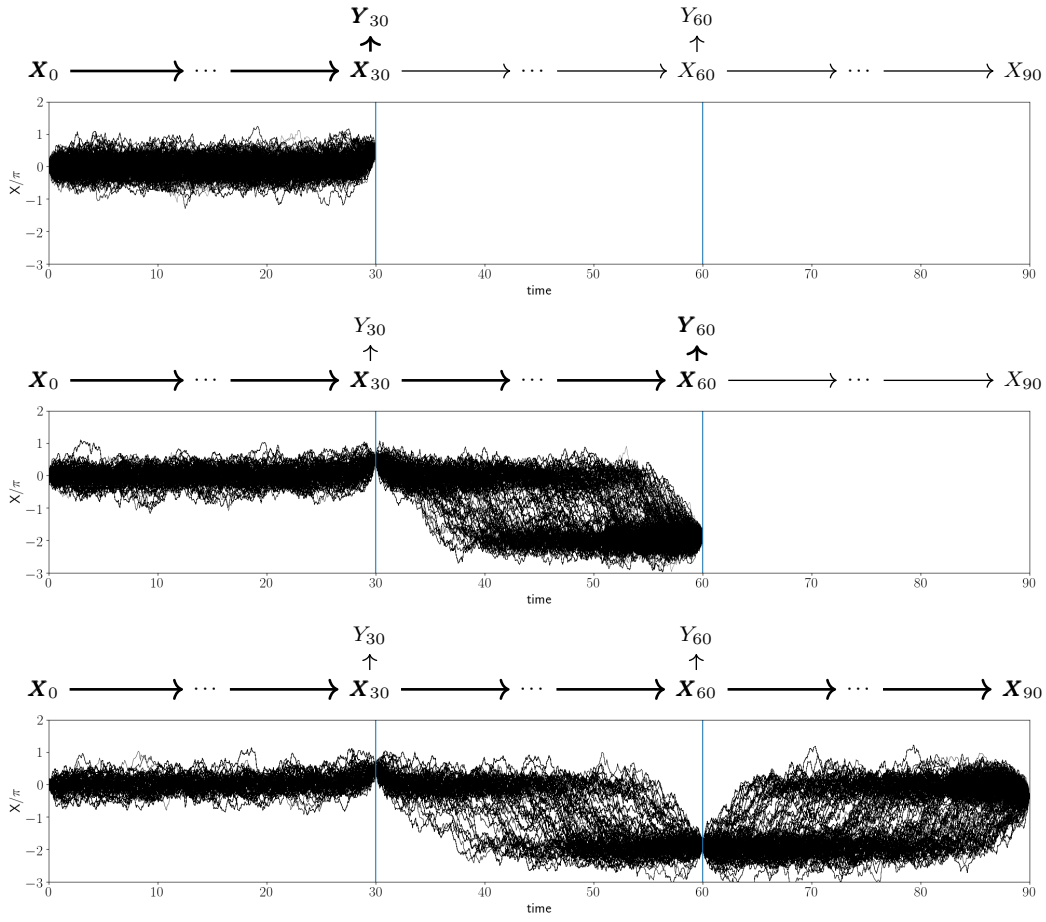


Figure 3.3: Illustration of the segmental sampling procedure.

In the first experiment, we set $X_0 = 0$, $Y_{30} = 1.49$, $Y_{60} = -5.91$ and $X_{90} = -1.17$. Note that the underlying stochastic process shows a jump behavior among the stable levels at $X_\lambda = 2k\pi,\ k = 0, \pm 1, \pm 2, \cdots$ (Lin et al., 2010). The four observations correspond to the

stable levels 0, 0, -2π and 0 accordingly. The process is likely to fluctuate around the stable level 0 during the first period. Then, it jumps to stable level -2π in the second period and eventually jumps back to stable level 0 in the third period.

Three levels of measurement errors in the observations Y_{30} and Y_{60} are investigated: $\sigma = 0.01$ for very accurate observations, $\sigma = 1.0$ for moderate accurate observations and $\sigma = 2.0$ for untrusted observations. Note that in this experiment we fix the observations Y_{30} and Y_{60} but change the underlying assumption of their distributions to reflect the strength and accuracy of the observations. A total number of $1,000$ forward paths are generated, and 300 backward pilots are used to estimate the resampling priority scores. Figure 3.4 plots the generated sample paths before weight adjustment for each level of error. Figure 3.5 shows the histogram of the marginal samples of $X_{60} = x_{60/\nu}$ before weight adjustment, which is obtained from the generated sample set $\{x_{0:T}^{(i)}\}_{i=1}^{n}$ without considering the weights.

It can be seen that when the observations are accurate ($\sigma = 0.01$), the two observations act like fixed-point constraints that force all sample paths to pass through the observations. When the observation error is large ($\sigma = 2$), a high proportion of sample paths remain at the original stable level while only a small proportion of paths are drawn towards the observations. The moderate error case ($\sigma = 1$) is a compromise between these two cases. The marginal distributions of X_{60} show clear differences in the above three cases. Samples from all three levels of error retain the jumping nature of underlying process and the SMCc-BP approach is capable of dealing with different levels of observational errors.

Next, we use the same setting as above except that setting $Y_{30} = 6.49$. Now the four observations $X_0 = 0$, $Y_{30} = 6.49$, $Y_{60} = -5.91$ and $X_{90} = -1.17$ correspond to the stable levels 0, 2π, -2π and 0, respectively. Since Y_{30} and Y_{60} differ by a gap of two stable levels, this is a very rare event. In this case, the Monte Carlo sample size is increased to 5,000 in order to overcome the degeneracy and to capture the rare event. Sample paths and histograms of the X_{60} samples before weight adjustment for different levels of error are shown in Figures 3.6 and Figures 3.7, respectively. In the large error case ($\sigma = 2$), most samples are concentrated around the stable level 0. As the error level decreases, the observation induced constraints become stronger, hence more sample paths are drawn towards the observations. Those figures provide the evidence that the priority scores estimated by the backward pilots are

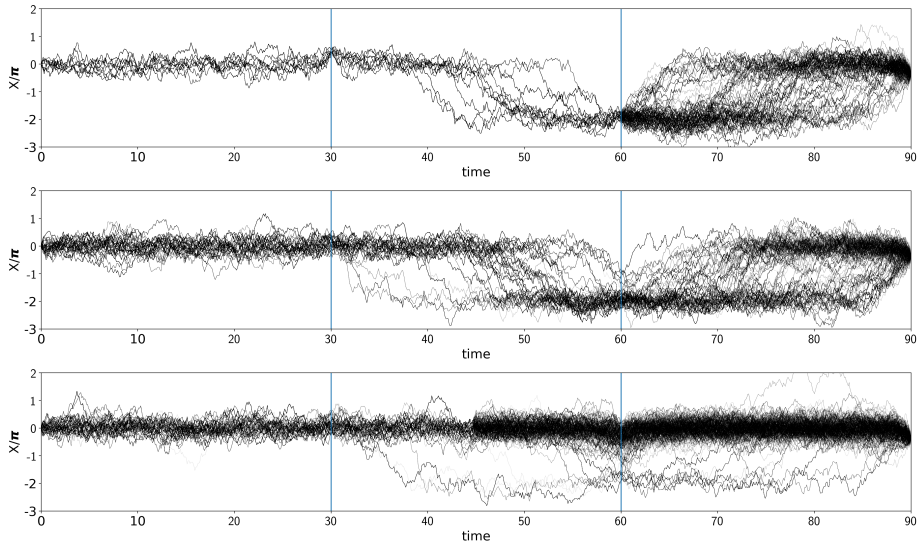


Figure 3.4: Sampled paths before weight adjustment for $\sigma = 0.01$ (top panel), $\sigma = 1.0$ (middle panel) and $\sigma = 2.0$ (bottom panel) when $X_0 = 0$, $Y_{30} = 1.49$, $Y_{60} = -5.91$ and $X_{90} = -1.17$

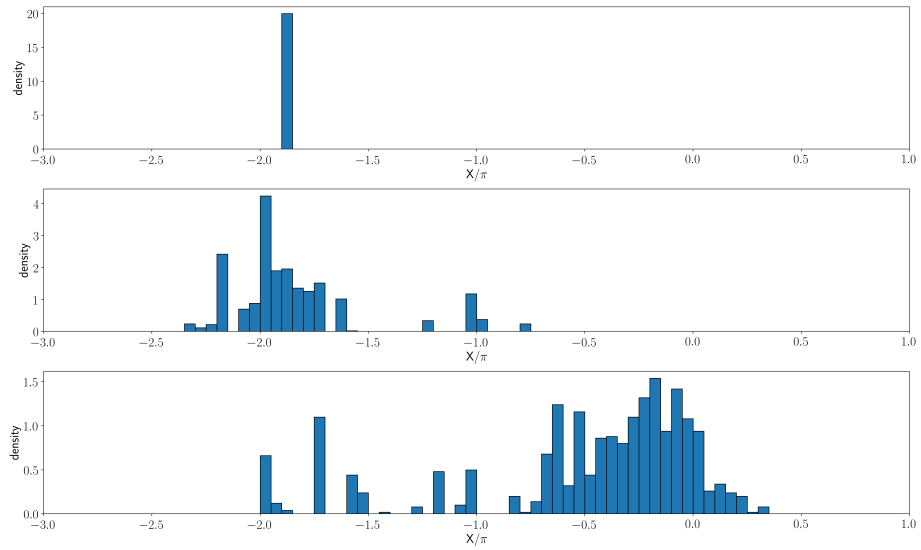


Figure 3.5: Histogram of the marginal samples of X_{60} before weight adjustment for $\sigma = 0.01$ (top panel), $\sigma = 1.0$ (middle panel) and $\sigma = 2.0$ (bottom panel) when $X_0 = 0$, $Y_{30} = 1.49$, $Y_{60} = -5.91$ and $X_{90} = -1.17$.

effective for different error levels under this extreme setting.

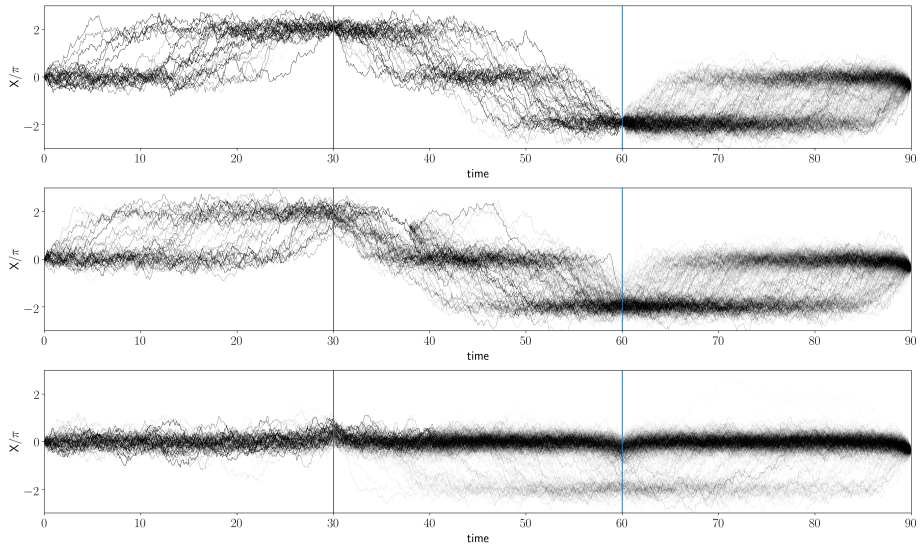


Figure 3.6: Sampled paths before weight adjustment for $\sigma = 0.01$ (top panel), $\sigma = 1.0$ (middle panel) and $\sigma = 2.0$ (bottom panel) when $X_0 = 0$, $Y_{30} = 6.49$, $Y_{60} = -5.91$ and $X_{90} = -1.17$.

3.6 Example: Optimal Trading Path

In asset portfolio management, the optimal trading path problem is a class of optimization problems which typically maximizes certain utility function of the trading path (Markowitz, 1959). The optimization problem is often complicated, especially when trading costs are considered. Kolm and Ritter (2015) turned such an optimization problem into a state space model and explored Monte Carlo methods to numerically solve it. Such a procedure is called state space emulation. More details on the emulation will be discussed in Chapter 4.

More specifically, let $x_{0:T} = (x_0, x_1, \ldots, x_T)$ be a trading path where x_t represents the holding position of an asset in shares at time t. In practice, a starting position x_0 and a target end position x_T are often imposed for optimal execution of a large order with minimum market impact. Without loss of generality, we impose two endpoints at $x_0 = 0$ and $x_T = 0$, respectively. Then it becomes an optimization problem to maximize the utility function

$$u(x_{0:T}) = -\sum_{t=1}^{T} c_t(x_t - x_{t-1}) - \sum_{t=1}^{T-1} h_t(y_t - x_t) \tag{3.12}$$

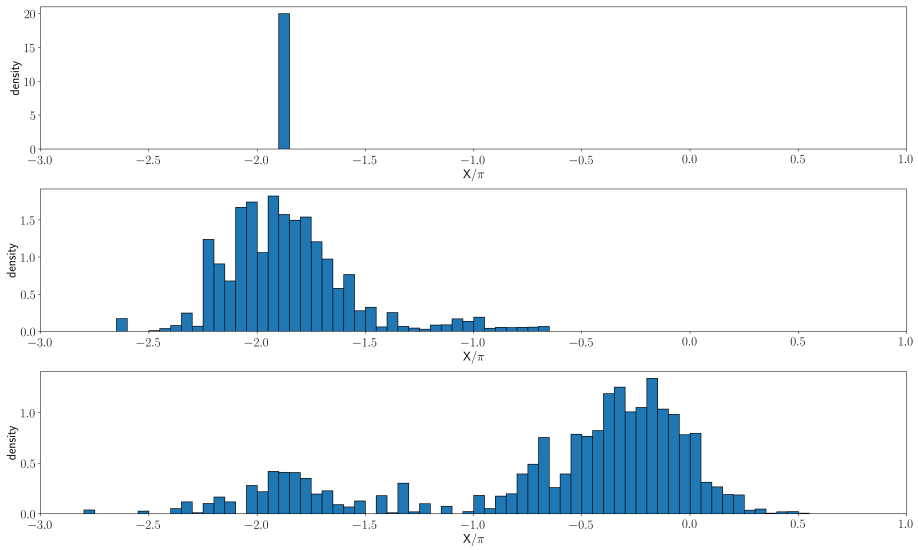


Figure 3.7: Histogram of the marginal samples of X_{60} before weight adjustment for $\sigma = 0.01$ (top panel), $\sigma = 1.0$ (middle panel) and $\sigma = 2.0$ (bottom panel) when $X_0 = 0$, $Y_{30} = 6.49$, $Y_{60} = -5.91$ and $X_{90} = -1.17$.

given $x_0 = 0$ and $x_T = 0$, where $(y_1, y_1, \ldots, y_{T-1})$ is a predetermined optimal trading path in an ideal world without trading costs, typically obtained by maximizing the risk-adjusted expected return under the Markowitz mean-variance theory (Markowitz, 1959).

Here $c_t(\cdot)$ is the trading cost function and $h_t(\cdot)$ stands for the utility loss due to the departure of the realized path from the ideal path. An emulating state space model can be implemented with the state equation $p(x_t \,|\, x_{t-1}) \propto \exp\{-c_t(x_t - x_{t-1})\}$ and the observation equation $p(y_t \,|\, x_t) \propto \exp\{-h_t(y_t - x_t)\}$. The joint posterior distribution of such a state space model is

$$\begin{aligned} p(x_{1:T-1} \mid x_0, y_{1:T-1}, x_T) &\propto \prod_{t=1}^{T} p(x_t \mid x_{t-1}) \prod_{t=1}^{T-1} p(y_t \mid x_t) \\ &\propto \exp\left\{ -\left[\sum_{t=1}^{T} c_t(x_t - x_{t-1}) + \sum_{t=1}^{T-1} h_t(y_t - x_t) \right] \right\}. \end{aligned} \tag{3.13}$$

Thus, it is a state space model with fixed endpoint constraints, which belongs to the multilevel constraint case, where the ideal path $y_{1:T}$ works as the periodic observations (weak constraint) and the holding position requirements at time 0 and T work as the strong constraints.

Following Kolm and Ritter (2015), we set $T = 20$. The ideal trading path is given directly

by

$$y_t = 25\exp\{-(t+1)/8\} - 40\exp\{-(t+1)/4\}.$$

The trading cost function and the utility loss due to tracking error are set to

$$c_t(x_t - x_{t-1}) = \frac{1}{2\sigma_x^2}\left[(x_t - x_{t-1})^2 + 2\alpha|x_t - x_{t-1}|\right] \quad \text{and} \quad h_t(y_t - x_t) = \frac{1}{2\sigma_y^2}(y_t - x_t)^2, \tag{3.14}$$

respectively, where $\sigma_x^2 = 0.25$ and $\sigma_y^2 = 1$. Here the trading cost is assumed to be a quadratic function of the trade size $|x_t - x_{t-1}|$, and α is a non-negative constant related to volatility and liquidity of the asset (Kyle and Obizhaeva, 2011), which we will specify in the following.

It can be seen that maximizing the utility function (3.12) is equivalent to find the maximize-a-posterior (MAP) path of distribution (3.13). We use the two-step method proposed in Godsill et al. (2001) to find the optimal trading path. First, we draw samples from the highly constrained conditional distribution (3.13) with the setting specified in (3.14). Then we discretize the space of x_t, $t = 1, \cdots, T-1$, based on the generated sample paths, and apply the Viterbi algorithm (Viterbi, 1967) to find an optimal path that maximizes the utility function (3.12) within the discretized state spaces. In general, the closer the generated sample paths are to the optimal one, the better trading path the Viterbi algorithm will produce. Section 5.4 will have a more detailed discussion on the Viterbi algorithm with SMC samples.

We investigate two cases of α in (3.14): $\alpha = 0$ and $\alpha = 0.5$. In both cases, we compare the performance of SMCc-BP with a standard SMC. The state equation $p(x_t \,|\, x_{t-1}) \propto \exp\{-c_t(x_t - x_{t-1})\}$ is used to generate forward paths in both methods. However, the standard SMC uses $\beta_t^{(i)} = w_t^{(i)}$ as the resampling priority scores, but SMCc-BP uses $\beta_t^{(i)} = w_t^{(i)}\widehat{p}(y_{t+1:T-1}, x_T \,|\, x_t^{(i)})$ estimated by the backward pilot method in Algorithm 6 for resampling, which takes the future information into account. The backward pilots are generated from the proposal distribution

$$r(\widetilde{x}_t|\widetilde{x}_{t+1}) \propto \exp\left\{-\frac{1}{2\sigma_x^2}\left[(\widetilde{x}_t - \widetilde{x}_{t+1})^2 + 2\alpha|\widetilde{x}_t - \widetilde{x}_{t+1}|\right]\right\}.$$

We use $m = 300$ backward pilots and generate $n = 2{,}000$ forward sample paths from

SMCc-BP. For the purpose of comparison, the standard SMC draws $n = 2,300$ forward paths such that both methods have a similar computational cost. In both methods, a resampling step is conducted when the ESS_t is less than $0.3n$.

3.6.1 Case 1: $\alpha = 0$

It can be seen that the state space model is linear and Gaussian when $\alpha = 0$. Hence, the Kalman filter (Kalman, 1960) can be applied to obtain an exact optimal solution. The sample paths generated from the standard SMC and SMCc-BP before weight adjustment, along with the exact optimal path and the 95% point-wise confidence intervals obtained by the Kalman filter are plotted in Figure 3.8. The samples from the standard SMC in the left panel have a much larger variance and most of them lie outside the 95% confidence region, while most samples from SMCc-BP in the right panel stay within the 95% confidence region. In SMCc-BP, the backward pilots bring the information about the future and guide the forward sample paths by resampling. On the other hand, without using any future information, the standard SMC sampler propagates blindly and suffers a large divergence between the sampling distribution and the target distribution at the end.

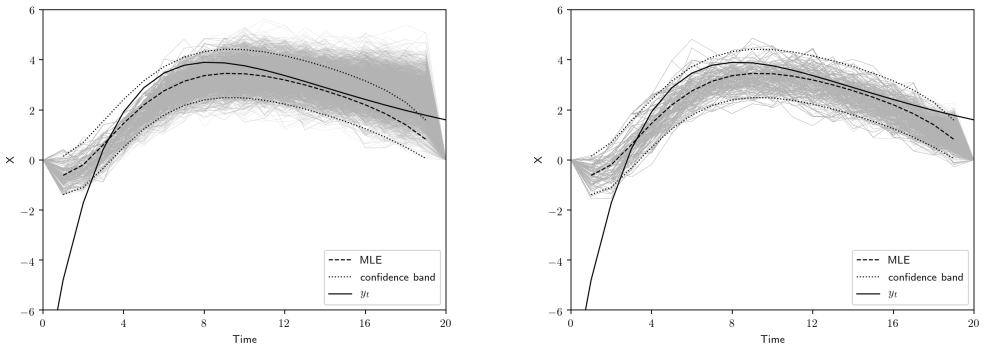


Figure 3.8: Sample paths from the standard SMC method (left panel) and from the SMCc-BP method (right panel) before weight adjustment when $\alpha = 0$.

Figure 3.9 shows the marginal densities of the samples generated by the standard SMC and SMCc-BP before weight adjustment (left column) and after weight adjustment (right column) at time $t = 4, 12, 19$. Both methods produce properly weighted samples, as the marginal densities for the samples after weight adjustment are close to the true one. How-

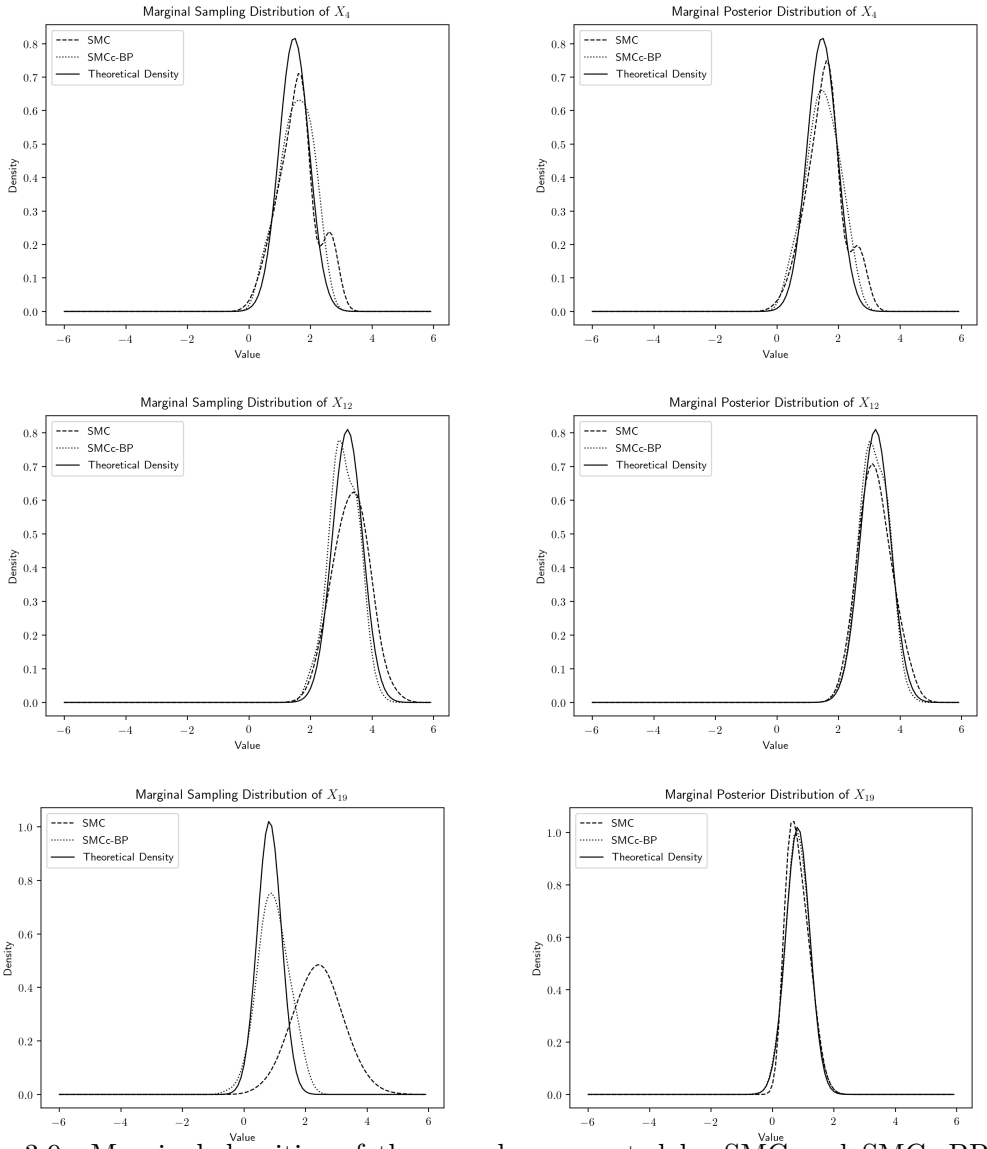


Figure 3.9: Marginal densities of the samples generated by SMC and SMCc-BP before weight adjustment (left column) and after weight adjustment (right column) at time $t = 4$ (row 1), $t = 12$ (row 2) and $t = 19$ (row 3) when $\alpha = 0$.

ever, the sampling distribution for x_{19} before weight adjustment under the standard SMC method has a large divergence from the true distribution, which results in a low efficiency for inference.

Figure 3.10 reports the mean squared errors (MSE) defined by

$$\mathrm{MSE}(t) = \frac{1}{L}\sum_{l=1}^{L}\left[\widehat{E}^{[l]}(x_t \,|\, x_0, y_{1:T-1}, x_T) - E(x_t \,|\, x_0, y_{1:T-1}, x_T)\right]^2,$$

where $E(x_t \,|\, x_0, y_{1:T-1}, x_T)$ is the true conditional mean obtained from the Kalman filter, and $\widehat{E}^{[l]}(x_t \,|\, x_0, y_{1:T-1}, x_T)$ is the conditional mean estimated by SMC or SMCc-BP in the l-th replication. We use $L = 1,000$ replications to compute the MSE's. It shows that in the period $8 \leqslant t \leqslant 17$ where the fixed points have limited impact, SMC and SMCc-BP have similar performance. But in the period $1 \leqslant t \leqslant 7$ where the observation y_t changes over time dramatically, SMCc-BP results in a smaller MSE than SMC as the future information is incorporated in its resampling step. In the period $t = 18$ and 19 where the end point constraint takes effect, the SMCc-BP approach also has a smaller MSE.

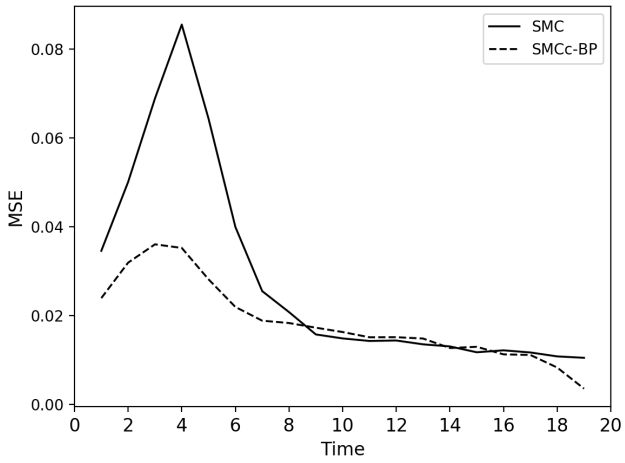


Figure 3.10: Mean squared error curves for SMC and SMCc-BP when $\alpha = 0$.

3.6.2 Case 2: $\alpha = 0.5$

When $\alpha = 0.5$, the state space model is non-Gaussian, hence there is no analytic solution to maximize the utility function (3.12). In this case, we run a standard SMC sampling with $n = 1,000,000$ sample paths to obtain the most likely sample path, the sample path with the largest likelihood value, together with 95% point-wise confidence intervals. The sample paths generated by SMC and SMCc-BP before weight adjustment, along with the most likely path and the 95% confidence region are plotted in Figure 3.11. Guided by the priority scores with future information, most samples generated by the SMCc-BP method stay within the 95% confidence region.

Figure 3.12 plots the marginal densities of the sample paths before weight adjustment

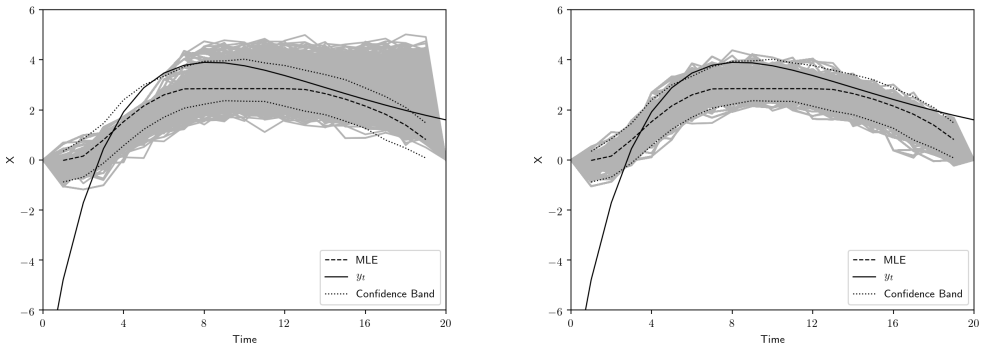


Figure 3.11: Sample paths from the standard SMC method (left panel) and from the SMCc-BP method (right panel) before weight adjustment when $\alpha = 0.5$.

(left column) and after weight adjustment (right column). The true marginal posterior distribution $p(x_t \mid x_0, y_{1:T-1}, x_T)$ is estimated from the same $n = 1{,}000{,}000$ SMC sample paths. At time $t = 19$, the distribution of SMCc-BP samples is much closer to the target one than that of SMC samples. Figure 3.13 plots the MSE's defined in (3.6.1). The results suggest that SMCc-BP reduces MSE at most times, especially in the periods $1 \leqslant t \leqslant 7$ and $13 \leqslant t \leqslant 19$.

3.6.3 Optimizing the Utility Function

The Viterbi algorithm (Viterbi, 1967) is a dynamic programming algorithm to find the most likely trajectory in a finite-state hidden Markov model. In this example, we discretize the state space based on the generated Monte Carlo state samples to utilize the Viterbi algorithm to find the optimal path and the optimal value of the utility function $u(x_{0:T})$ in (3.12). Specifically, given $\mathcal{X}_t = \{x_t^{(i)}\}_{i=1,\cdots,n}$ being the collection of samples of x_t generated by SMC or SMCc-BP, the optimal path $(\hat{x}_1, \cdots, \hat{x}_{T-1})$ is found by solving the following optimization problem

$$(\hat{x}_1, \cdots, \hat{x}_{T-1}) = \underset{x_1 \in \mathcal{X}_1, \ldots, x_{T-1} \in \mathcal{X}_{T-1}}{\arg\max} u(x_{0:T})$$

with the Viterbi algorithm.

In this experiment, we use $m = 300$ backward pilots and generate $n = 500$ Monte Carlo forward samples from SMCc-BP. For comparison, $n = 800$ samples are generated from the

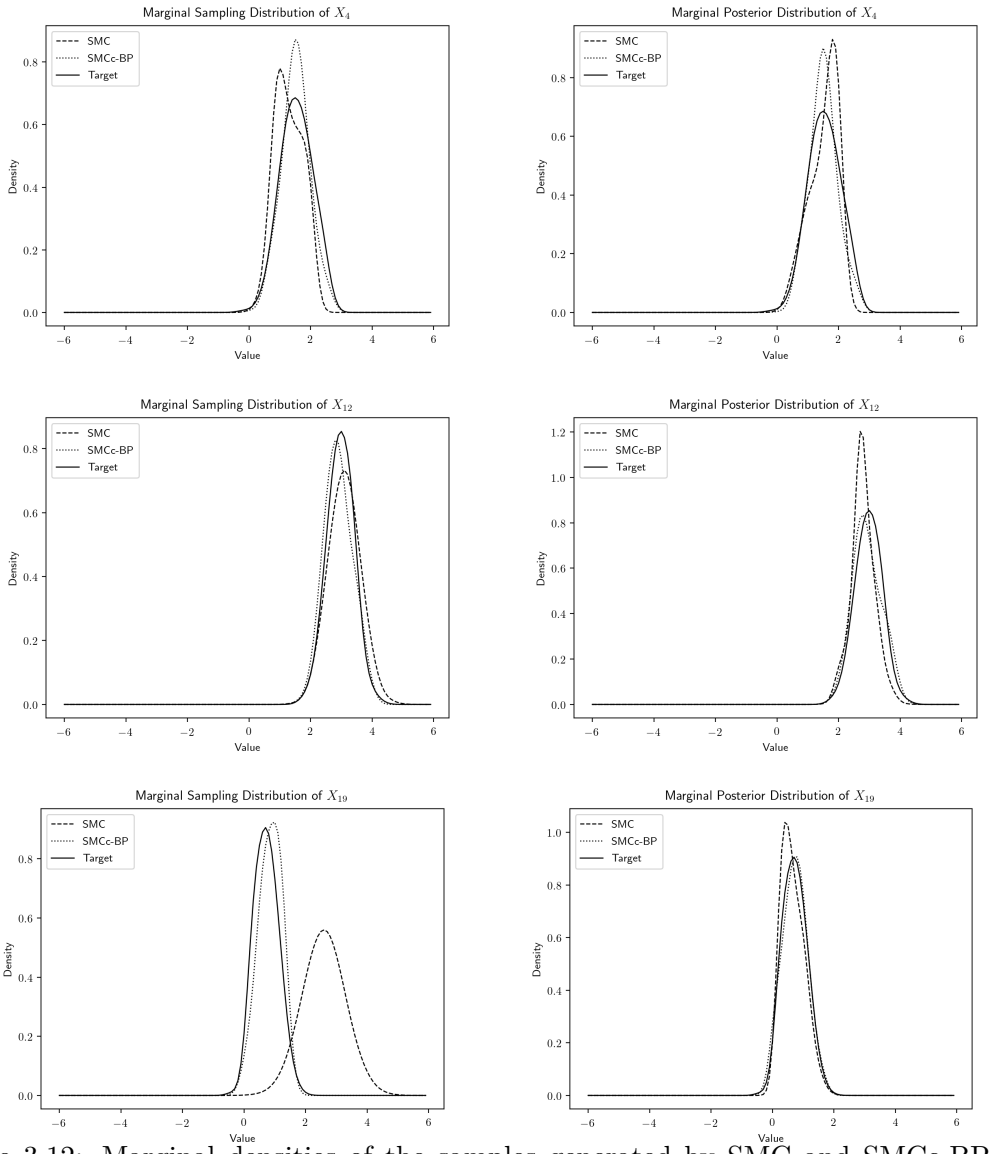


Figure 3.12: Marginal densities of the samples generated by SMC and SMCc-BP before weight adjustment (left column) and after weight adjustment (right column) at time $t = 4$ (row 1), $t = 12$ (row 2) and $t = 19$ (row 3) when $\alpha = 0.5$.

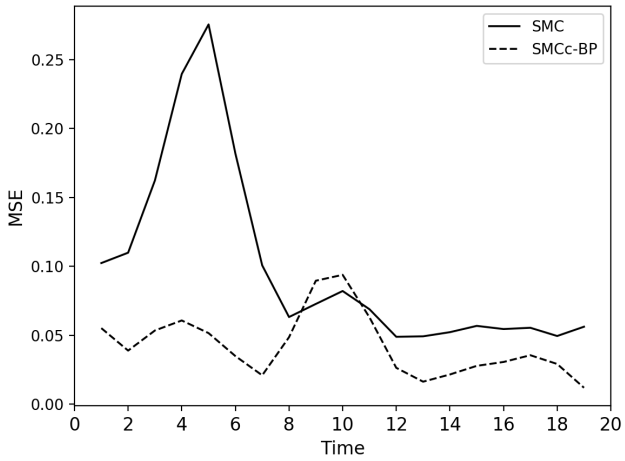


Figure 3.13: Mean squared error curves for SMC and SMCc-BP when $\alpha = 0.5$.

standard SMC method. The experiment is replicated 1,000 times. The optimal values of the utility function (up to a constant) solved by the Viterbi algorithm based on SMC samples and SMCc-BP samples respectively are reported in the boxplots in Figure 3.14. The true optimal value is marked by the horizontal lines. When $\alpha = 0.0$, the true optimal value is obtained by the Kalman filter. When $\alpha = 0.5$, the "true" optimal value is computed by the Viterbi algorithm based on a large number ($n = 10,000$) of SMC samples. Compared to the standard SMC method, the SMCc-BP method generates more samples around the true optimal path in the same amount of computation time by incorporating future information through resampling, hence it creates a better discrete state space for the Viterbi algorithm. As a result, the Viterbi algorithm based on SMCc-BP samples can produce trading paths with larger utility function values for both $\alpha = 0$ and $\alpha = 0.5$ cases.

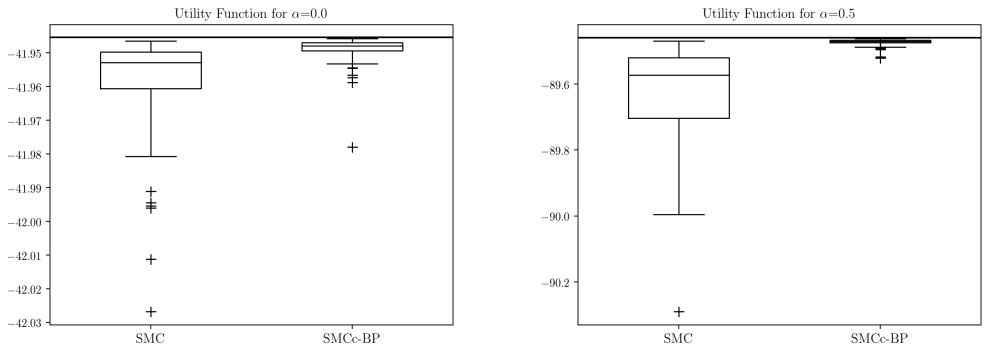


Figure 3.14: Boxplots of optimal values of utility function (3.12) solved by the Viterbi algorithm based on SMC samples and SMCc-BP samples when $\alpha = 0$ (left panel) and $\alpha = 0.5$ (right panel). The horizontal lines are the true optimal values.

CHAPTER 4

State Space Emulation

High dimensional global optimization algorithms are being widely investigated since more and more applications involve high dimensional complex data nowadays. The gradient descent algorithm and its variations (Bertsekas, 1997) require the objective function to be convex or uni-modal so that the found local optimal is global. Recent research in machine learning involves many non-convex optimization problems (Anandkumar et al., 2014; Arora et al., 2012; Netrapalli et al., 2014; Agarwal et al., 2014). However, many non-convex problems remain NP-hard and the theory is only available for their convex relaxations (Jain et al., 2017). Deterministic optimization algorithms (Hooke and Jeeves, 1961; Nelder and Mead, 1965; Land and Doig, 1960) may result in certain types of exhaustive search, which is computationally expensive in a high dimensional space. As an alternative, stochastic optimization algorithms utilize Monte Carlo simulations to explore the parameter space in a stochastic and often more efficient way (Kiefer et al., 1952; Kirkpatrick et al., 1983; Mei et al., 2018).

In this chapter, we focus on an emulation approach, which reformulates a high dimensional optimization problem into the problem of finding the most likely state path problem in a state space model. The most likely path problem as discussed in Section 2.1.3 is to equivalent to a high-dimensional optimization problem that maximizes the posterior (2.7). The emulation does the reverse: an optimization problem is rewritten in an equivalent form of (2.7). The optimization problem is then solved under the emulated state space model based on the Monte Carlo samples drawn with SMC techniques.

This chapter will mainly focus on transforming high-dimensional optimization problems

into proper state space models. The proposed SMC method (annealed SMC) to solve the most likely path problem for a state space model is postponed to next chapter.

4.1 Previous Work on Emulation

There exist several heuristic approaches that use the idea of emulation.

Cai et al. (2009) considers the variable selection problem in high dimensional regression analysis. Each of the p-dimensional variables $x_1,\ldots,x_p$ takes value in $\{0,1\}$, where $x_k = 1$ indicates the $k-th$ covariate is included in the regression model. A variable selection problem is now equivalent to optimize a criterion function over $(x_1,\ldots,x_p) \in \{0,1\}^p$. Exhaust search of all 2^p possibilities is expense. Cai et al. (2009) reconstructed the objective function to the posterior function of a state space model by setting $x_1,\ldots,x_p$ as the latent variables and the observations as $y_{1:p}$, even though the p covariates have no chronological order in nature. Cai et al. (2009)'s algorithm in finding the most likely path is a combination of sequential Monte Carlo and dynamic programming, which strongly depends on the fact that the common support of all latent variables, $\{0,1\}$, is finite. Therefore, it is difficult to generalize the algorithm to a continuous space.

As mentioned in the example of Section 3.6, the utility function of a portfolio optimization problem is reformulated such that the emulated state space model holds an equivalent likelihood/posterior function and therefore can be solved under the context of the state space model. Such a transformation was first proposed by Kolm and Ritter (2015) and in their original paper, a Viterbi algorithm is applied to the Monte Carlo samples to find the optimal trading path as shown in Section 3.6. Similarly, Irie and West (2016) relates the multi-period portfolio optimization problem to the log-likelihood of a mixture of linear Gaussian dynamic systems such that the Kalman filter (Kalman, 1960) is used to find the optimal solution of each component system and EM algorithm (Dempster et al., 1977) is applied to find the most likely path of the overall emulated state space model. In above two emulation works, the Viterbi algorithm requires the dynamic system to be Markovian and non-singular. And the combination of the Kalman filter and EM algorithm proposed in Irie and West (2016) works only when the underlying distribution can be represented as

a mixture of Gaussian distributions.

4.2 Principle of Emulation

Suppose the original optimization problem is

$$\min_{x \in \mathcal{X}^d} \ f(x),$$

where $f : \mathcal{X}^d \to \mathbb{R}$ is the objective function to be minimized. Let $\xi : \mathbb{R} \to [0, +\infty)$ be a monotone decreasing function. Then minimizing $f(x)$ is equivalent to maximizing $\phi(x) := \xi(f(x))$ such that

$$\arg\min_{x \in \mathcal{X}^d} \ f(x) = \arg\max_{x \in \mathcal{X}^d} \ \phi(x),$$

when the "argmin" exists and is unique.

Furthermore, if there exists a state space model whose posterior function (2.7) is proportional to $\phi(x)$ such that

$$\pi(x_{1:T} \mid y_{1:T}) \propto \phi(x_{1:T}) = \xi(f(x_{1:T})) \tag{4.1}$$

with artificially designed state equations $\{f_t(\cdot)\}_{t=1}^T$ and observation equations $\{g_t(\cdot)\}_{t=1}^T$, we call the state space model an "emulated" state space model. The observations $y_{1:T}$ can be either certain observations involving in the original optimization problem (e.g. the observed points in the smoothing spline problem in Section 4.3.1) or artificially designed. Note that, in (4.1), we ignore x_0 and set $T = d$ even when the index does not have any chronological meaning.

When $\phi(\cdot)$ is integrable with respect to Lebesgue measure on $\mathcal{X}^T$ and bounded, there always exist a trivial emulated state space model such that

$$\phi(x_{1:T}) = \phi_1(x_1) \prod_{t=2}^{T} \phi_t(x_t \mid x_{1:t-1}),$$

where

$$\phi_1(x_1) = \int_{\mathcal{X}^{T-1}} \phi(x_{1:T}) dx_2 \cdots dx_T$$

and

$$\phi_t(x_t \mid x_{1:t-1}) = \frac{\int_{\mathcal{X}^{T-t}} \phi(x_{1:T}) dx_{t+1} \cdots dx_T}{\int_{\mathcal{X}^{T-t+1}} \phi(x_{1:T}) dx_t \cdots dx_T}.$$

Often such a series of conditional distribution is difficult to sample from or to be evaluated. However, in certain problems as our examples shown later, it is possible to reformulate the conditional distribution to $\phi_t(x_t \mid x_{1:t-1}) = f_t(x_t \mid x_{1:t-1}) g_t(y_t \mid x_t)$, in which $f_t(x_t \mid x_{1:t-1})$ is easy to generate samples from and $g_t(y_t \mid x_t)$ is easy to be evaluated, for some designed y_t. Minimizing the objective function f is then the same as finding the most likely path for the emulated state space model.

A common choice for the function $\xi(\cdot)$ is the Boltzmann distribution function

$$\xi(s) = e^{-\kappa s}, \tag{4.2}$$

where κ is a positive constant that relates to the temperature in statistical physics. In statistics, the Boltzmann function in (4.2) links the least square method to the maximum likelihood approach with i.i.d. Gaussian noise. In addition, with this choice of $\xi(\cdot)$, the system has a physical interpretation: The objective function $f(\cdot)$ is regarded as the possible energy levels in a non-quantum thermodynamic system. Assuming no interactions, the number of particles at the energy $f(x)$ follows the Boltzmann distribution under thermodynamic equilibrium. The integrability of $\phi(x)$ ensures the existence of the canonical partition function such that this physical canonical system is valid. The minimization of $f(\cdot)$ is now equivalent to find the base energy level, which inspires the use of simulated annealing of this thermodynamic system. More details will be discussed in Chapter 5.

4.3 Emulation Examples

4.3.1 Cubic Smoothing Spline

Consider a nonparametric regression model

$$y_t = m(x_t) + \epsilon_t$$

with equally spaced x_t. Without loss of generality, let $x_t = t$ and treat it as time.

The cubic smoothing spline method (Green and Silverman, 1993) estimates a continuous function $m(t)$ by minimizing

$$\sum_{t=1}^{T}(y_t - m(t))^2 + \lambda \int \left[m''(t)\right]^2 dt. \tag{4.3}$$

The first term in (4.3) is the total squared tracking errors at the observation times and the second term is the penalty term on the smoothness of the latent function $m(\cdot)$, where λ controls the regularization strength. Given the values of $m(1),\ldots,m(T)$, the minimizer of the second term is a natural cubic spline that interpolates $m(1),\ldots,m(T)$ (see Green and Silverman (1993)). Hence, the solution to minimize (4.3) is a natural cubic spline, which is second-order continuously differentiable and is a cubic polynomial in all intervals $[t,t+1]$ for $t = 1,\ldots,T-1$ and is linear outside $[1,T]$.

Define the derivatives of $m(t)$ at each observation at time t as

$$a_t = m(t), \quad b_t = m'(t), \quad c_t = m''(t)/2, \quad d_t = \lim_{s\to t_-} m'''(s)/6.$$

By the constraints of natural cubic spline, we have the following recursive relationships:

$$\begin{aligned} a_{t+1} &= a_t + b_t + c_t + d_{t+1}, \\ b_{t+1} &= b_t + 2c_t + 3d_{t+1}, \\ c_{t+1} &= c_t + 3d_{t+1}, \end{aligned}$$

with $c_1 = c_T = 0$. Furthermore, by substituting d_{t+1} with $(c_{t+1} - c_t)/3$ in the expressions of a_t and b_t, we have

$$a_{t+1} = a_t + b_t + (c_{t+1} + 2c_t)/3, \tag{4.4}$$

$$b_{t+1} = b_t + c_t + c_{t+1}. \tag{4.5}$$

We will use the recursive relationships in (4.4) and (4.5) for the construction of state space

emulation. With this notation, the second term in (4.3) can be expended as

$$\lambda \int \left[m''(t)\right]^2 dt = \lambda \sum_{t=1}^{T-1} \int_t^{t+1} \left[6(s-t)d_{t+1} + 2c_t\right]^2 ds = \frac{4}{3}\lambda \sum_{t=1}^{T-1} (c_t^2 + c_t c_{t+1} + c_{t+1}^2).$$

In this case, the original optimization problem (4.3) over all second order differentiable functions becomes minimizing

$$f(x_{1:T}) = \sum_{t=1}^{T} (y_t - a_t)^2 + \frac{4}{3}\lambda \sum_{t=1}^{T-1} (c_t^2 + c_t c_{t+1} + c_{t+1}^2), \tag{4.6}$$

where $x_{1:T} = \{(a_t, b_t, c_t)\}_{t=1,\ldots,T}$ satisfies the recursive relationships (4.4) and (4.5) and the boundary condition $c_1 = c_T = 0$. Note that $x_{1:T}$ completely defines the cubic smoothing spline solution $\hat{m}(t)$.

With a positive inverted temperature κ, an emulated state space model is one such that whose likelihood of $x_{1:T}$ conditioned on $y_1, \ldots, y_T$ is $\pi(x_{1:T} \mid y_{1:T}) \propto e^{-\kappa f(x_{1:T})}$, with $f(\cdot)$ defined in (4.6). One possible way to decompose $\pi(x_{1:T} \mid y_{1:T})$ into the likelihood of a state space model is the following.

$$\begin{aligned}
\pi(x_{1:T} \mid y_{1:T}) &\propto \exp\left(-\kappa f(x_{1:T})\right) \\
&= \exp\left(-\kappa \sum_{t=1}^{T} (y_t - a_t)^2 - \frac{4\lambda\kappa}{3} \left(\sum_{t=1}^{T-1} (c_t^2 + c_t c_{t+1} + c_{t+1}^2)\right)\right) \\
&= \left(\prod_{t=1}^{T} e^{-\kappa (y_t - a_t)^2}\right) \left(\prod_{t=2}^{T} e^{-\frac{2\lambda\kappa}{3(2-\sqrt{3})} (c_t + (2-\sqrt{3})c_{t-1})^2}\right),
\end{aligned} \tag{4.7}$$

where κ, the "temperature" parameter, controls the shape of the distribution.

The second term of (4.7) provides a construction of a first order vector auto-regressive process on $\{x_t = (a_t, b_t, c_t)\}_{t=1,\ldots,T}$ as the state equation

$$\begin{bmatrix} a_t \\ b_t \\ c_t \end{bmatrix} = \begin{bmatrix} 1 & 1 & \sqrt{3}/3 \\ 0 & 1 & \sqrt{3}-1 \\ 0 & 0 & -(2-\sqrt{3}) \end{bmatrix} \begin{bmatrix} a_{t-1} \\ b_{t-1} \\ c_{t-1} \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1 \\ 1 \end{bmatrix} \eta_t, \tag{4.8}$$

with $\eta_t \sim \mathcal{N}(0, \sigma_b^2)$, $\sigma_b^2 = \dfrac{3(2-\sqrt{3})}{4\lambda\kappa}$. The first term of (4.7) provides the construction of the observation equation

$$y_t = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_t \\ b_t \\ c_t \end{bmatrix} + \varepsilon_t, \tag{4.9}$$

with $\varepsilon_t \sim \mathcal{N}(0, \sigma_y^2)$, $\sigma_y^2 = 1/(2\kappa)$, and the initial values

$$a_1 \sim \mathcal{N}(y_1, \sigma_y^2),\ b_1 \sim 1 \text{ and } c_1 = 0.$$

4.3.2 Regularized Linear Regression

LASSO (Tibshirani, 1996) is a widely-used regularized linear regression estimation procedure that can perform variable selection and parameter estimation at the same time.

Consider the regression model

$$\boldsymbol{Y} = \sum_{j=1}^{p} \beta_j \boldsymbol{Z}_j + \boldsymbol{\eta}$$

where $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_p \in \mathbb{R}^n$ are the p covariates that are used to model the dependent variable $\boldsymbol{Y} \in \mathbb{R}^n$ and $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma_y^2 I_n)$. A LASSO estimator of $(\beta_1, \ldots, \beta_p)$ is the minimizer of

$$f(\beta_1, \ldots, \beta_p) = \|\boldsymbol{Y} - \beta_1 \boldsymbol{Z}_1 - \cdots - \beta_p \boldsymbol{Z}_p\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{4.10}$$

For a fixed set of $(\beta_1, \ldots, \beta_p)$, for $t = 1, \ldots, p$, define the partial residual $\boldsymbol{\epsilon}_t$ as

$$\boldsymbol{\epsilon}_t = \boldsymbol{Y} - \sum_{l=1}^{t} \beta_l \boldsymbol{Z}_l \tag{4.11}$$

and $\boldsymbol{\epsilon}_0 = \boldsymbol{Y}$. Since

$$\|\boldsymbol{\epsilon}_t\|_2^2 = \|\boldsymbol{\epsilon}_{t-1} - \beta_t \boldsymbol{Z}_t\|_2^2 = \|\boldsymbol{\epsilon}_{t-1}\|_2^2 + \|\boldsymbol{Z}_t\|_2^2 \left(\beta_t - \frac{\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t}{\|\boldsymbol{X}_t\|_2^2} \right)^2 - \frac{(\boldsymbol{\epsilon}_{t-1}' \boldsymbol{X}_t)^2}{\|\boldsymbol{Z}_j\|_2^2},$$

we have

$$f(\beta_1,\ldots,\beta_p) = \|\boldsymbol{\epsilon}_p\|_2^2 + \lambda\sum_{t=1}^{p}|\beta_t| = \|\boldsymbol{Y}\|_2^2 + \sum_{t=1}^{p}\left\{\|\boldsymbol{Z}_t\|_2^2\left(\beta_t - \frac{\boldsymbol{\epsilon}_{t-1}'\boldsymbol{Z}_t}{\|\boldsymbol{Z}_t\|_2^2}\right)^2 - \frac{(\boldsymbol{\epsilon}_{t-1}'\boldsymbol{Z}_t)^2}{\|\boldsymbol{Z}_t\|_2^2} + \lambda|\beta_t|\right\}. \tag{4.12}$$

Let $x_t = \beta_t$ and $x_{1:t} = (\beta_1,\ldots,\beta_t)$. An emulated state space model can be designed so that

$$\begin{aligned}\pi(x_{1:p}) &\propto \exp\{-\kappa f(x_{1:p})\} \\ &\propto \prod_{t=1}^{p}\exp\left\{-\kappa\|\boldsymbol{Z}_t\|_2^2\left(x_t - \frac{\boldsymbol{\epsilon}_{t-1}'\boldsymbol{Z}_t}{\|\boldsymbol{Z}_t\|_2^2}\right)^2\right\} \times \prod_{t=1}^{p}\exp\left\{-\kappa\lambda|x_t| + \kappa\frac{(\boldsymbol{\epsilon}_{t-1}'\boldsymbol{Z}_t)^2}{\|\boldsymbol{Z}_t\|_2^2}\right\}. \end{aligned} \tag{4.13}$$

The first term of (4.13) leads to the state equation

$$f_t(x_t \mid x_{1:t-1}) \propto \exp\left\{-\kappa\|\boldsymbol{Z}_t\|_2^2\left(x_t - \frac{\boldsymbol{\epsilon}_{t-1}'\boldsymbol{Z}_t}{\|\boldsymbol{Z}_t\|_2^2}\right)^2\right\}, \tag{4.14}$$

and the second term leads to the observation equation

$$g_t(w_t \mid x_t) \propto \alpha_t \exp\{-\alpha_t w_t\}, \tag{4.15}$$

where

$$\alpha_t = \exp\left\{-\kappa\lambda|x_t| + \kappa\frac{(\boldsymbol{\epsilon}_{t-1}'\boldsymbol{Z}_t)^2}{\|\boldsymbol{Z}_t\|_2^2}\right\},$$

with observation $w_t = 0$ for all t.

Note that $\boldsymbol{\epsilon}_{t-1}$ is a function of $x_{1:t-1}$ as defined in (4.11) and is available at time t. The observation equation g_t and the observation value $w_t = 0$ are imposed to incorporate α_t in $\pi(x_{1:p})$. The emulation for LASSO can be extended to other penalized regression with different penalty terms by changing α_t accordingly.

4.3.3 Optimal Trading Path

Here we recap the optimal trading path problem demonstrated in Section 3.6.

Specifically, the optimal trading path problem is a class of optimization problems which typically maximizes certain utility function of the trading path (Markowitz, 1959). Kolm and Ritter (2015) and Irie and West (2016) proposed to turn such an optimization problem

to an emulated state space model. To be more specific, let $x_{0:T} = (x_0, \ldots, x_T)$ be a trading path in which x_t represents the position held at time t. Kolm and Ritter (2015) propose to maximize the following utility function.

$$u(x_{0:T}) = -\sum_{t=1}^{T} c_t(x_t - x_{t-1}) - \sum_{t=0}^{T} h_t(y_t - x_t), \tag{4.16}$$

where $(y_0, \ldots, y_T)$ is a predetermined optimal trading path in an ideal world without trading costs, typically obtained by maximizing the risk-adjusted expected return under the Markowitz mean-variance theory (Markowitz, 1959). Kolm and Ritter (2015) provides a construction of $(y_0, \ldots, y_T)$ based on the term structure of the underlying asset's *alpha* (the excess expected return relative to the market). Let $c_t(\cdot)$ represent the transaction cost which is often assumed to be a quadratic function of the absolute position change $|x_t - x_{t-1}|$. Without loss of generality, we parametrize it as

$$c_t(|x_t - x_{t-1}|) = \frac{1}{2\sigma_x^2}\left(|x_t - x_{t-1}|^2 + 2\alpha|x_t - x_{t-1}| + \alpha^2\right),$$

where α is a non-negative constant related to the volatility and liquidity of the asset (Kyle and Obizhaeva, 2011). Let $h_t(\cdot)$ be the utility loss due to the departure of the realized path from the ideal path. We use the squared loss

$$h_t(y_t - x_t) = \frac{1}{2\sigma_y^2}(y_t - x_t)^2.$$

Then the objective function is

$$\pi(x_{0:T} \mid y_{1:T}) \propto e^{-\kappa u(x_{1:T})} \propto \prod_{t=1}^{T} \exp\left(-\frac{\kappa(|x_t - x_{t-1}| + \alpha)^2}{2\sigma_x^2}\right) \prod_{t=1}^{T} \exp\left(-\frac{\kappa(y_t - x_t)^2}{2\sigma_y^2}\right).$$

Taking the position constraint $x_0 = x_T = 0$ into consideration as discussed in Section 3.6, an emulated state space model can therefore be constructed as

$$f_t(x_t \mid x_{t-1}) \propto \exp\left(-\frac{\kappa(|x_t - x_{t-1}| + \alpha)^2}{2\sigma_x^2}\right), \tag{4.17}$$

$$g_t(y_t \mid x_t) \propto \exp\left(-\frac{\kappa(y_t - x_t)^2}{2\sigma_y^2}\right). \tag{4.18}$$

With the state equation (4.17) and the observation equation (4.18), the corresponding state space model has a likelihood function proportional to $\exp(-\kappa u(x_{1:T}))$.

4.3.4 L1 Trend Filtering

L1 trend filtering (Kim et al., 2009) is a variation of the Hodrick-Prescott filtering (Hodrick and Prescott, 1997). An $\ell 1$ trend filtering on $y_1, \ldots, y_T$ is defined to be the minimizer of the objective function

$$f(x_1, \ldots, x_T) = \sum_{t=1}^{T} (Y_t - x_t)^2 + \lambda \sum_{t=2}^{T-1} |x_{t-1} - 2x_t + x_{t+1}|. \tag{4.19}$$

Minimizing (4.19) tends to produce a piece-wise linear function due to the ℓ_1 penalty on second-order difference. An emulated state space model is designed to have the following Boltzmann likelihood function.

$$\pi(x_{1:T}) \propto e^{-\kappa f(x_{1:T})/2} = \prod_{t=1}^{T} \exp\left\{-\frac{\kappa}{2}(y_t - x_t)^2\right\} \prod_{t=3}^{T} \exp\left\{-\frac{\kappa}{2\lambda}|x_t - (2x_{t-1} - x_{t-2})|\right\}. \tag{4.20}$$

The first term of (4.20) leads to the observation equation

$$y_t = x_t + \epsilon_t, \tag{4.21}$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma_y^2)$ with $\sigma_y^2 = 1/\kappa$. The second term of (4.20) leads to the following second order auto-regressive process of the states

$$x_t = 2x_{t-1} - x_{t-2} + \eta_t, \tag{4.22}$$

where $\eta_t \sim Laplace(0, \lambda_x)$ with $\lambda_x = 2/(\lambda\kappa)$.

CHAPTER 5

Annealed Sequential Monte Carlo

5.1 Most Likely Path

The most likely path problem is to find the optimal latent variables $x^*_{1:T}$ that maximizes the posterior function $\pi(x_{1:T} \mid y_{1:T})$ of a state space model defined in (2.7). Particularly, we define

$$x^*_{1:T} = \underset{x_{1:T} \in \mathcal{X}^T}{\arg\max}\ \pi(x_{1:T} \mid y_{1:T}), \tag{5.1}$$

where $\mathcal{X}$ is the common support for the states $x_t, t = 1, \ldots, T$. When the state space model is constructed to emulate a certain objective function $f(\cdot)$ as discussed in Chapter 4, the most likely path $x^*_{1:T}$ is also the optimal of $f(\cdot)$ as ensured by (4.1).

Instead of solving (5.1) directly, which trace back to the original optimization of $f(\cdot)$, numerical solution to (5.1) can be carried out based on a set of (sequential) Monte Carlo samples. Specifically, let $\{(x^{(i)}_{1:T}, w^{(i)})\}_{i=1}^N$ be a set of Monte Carlo samples, which are properly weighted with respect to the posterior distribution $\pi(x_{1:T} \mid y_{1:T})$.

A natural and easy way to estimate $x^*_{1:T}$ in (5.1) is to use the empirical maximum-a-likelihood (MAP) path such that

$$\hat{x}^{(map)}_{1:T} = \underset{x_{1:T} \in \{x^{(i)}_{1:T}\}_{i=1}^N}{\arg\max}\ \pi(x_{1:T} \mid y_{1:T}). \tag{5.2}$$

In (5.2), the posterior function is optimized over the finite set of N sample paths. However, in order to control the accuracy of $\hat{x}^{(map)}_{1:T}$, a large number of sample paths is required when the dimension T is high. To illustrate such a curse of dimensionality, we assume the desired

estimate of the optimal point should satisfy

$$\|\hat{x}_{1:T}^{(map)} - x_{1:T}^*\|_\infty < \epsilon,$$

for some constant ϵ. In other words, $\hat{x}_{1:T}^{(map)}$ lies in a T-dimensional cube with edge length 2ϵ centered at $x_{1:T}^*$. To ensure at least one sample path is expected to drop within this T-dimensional cube, the desired number of samples is approximately

$$N^* \propto \left(\frac{|\mathcal{X}|}{\epsilon}\right)^T,$$

where $|\mathcal{X}|$ is the volume of $\mathcal{X}$. As a result, as the dimension T increase, N^* increases exponentially in order to achieve a similar accuracy.

The problem of (5.2) is that N samples $\{x_t^{(i)}\}_{i=1}^N$ are generated for every time t. There exist in total N^T combinations of the tuple $(x_1, \cdots, x_T)$. But (5.2) only considers N of N^T combinations — a minor portion of all possibilities from the Monte Carlo samples. A generalized version is to optimize the posterior function over these N^T combinations as shown in the following.

$$\hat{x}_{1:T}^{(joint)} = \underset{\{x_{1:T}: x_t \in \{x_t^{(i)}\}_{i=1}^N\}}{\arg\max} \pi(x_{1:T} \mid y_{1:T}). \tag{5.3}$$

The accuracy can be improved as the optimization (5.3) is over a superset of that one of (5.2). The concern is the computational cost as much more possibilities are considered in (5.3). However, when the emulated state space model is Markovian and non-singular, the optimization problem in (5.3) can be solved in a dynamic programming way with a linear computational time. The algorithm is known as the Viterbi algorithm (Viterbi, 1967). Details of this algorithm will be discussed later in Section 5.4. Similar to the analysis of required number of samples for the map estimator, the desired number of samples for the estimator in (5.3) is now

$$N^* \propto \frac{|\mathcal{X}|}{\epsilon},$$

by assuming $x_t^{(i)}, i = 1, \ldots, N$ are i.i.d. and are independent with $x_s^{(j)}, j = 1, \ldots, N$, $s \neq t$.

It is obvious that to achieve a similar accuracy, the estimator (5.3) requires a much less number of samples compared with the one in (5.2).

Aforementioned approaches assumes that the Monte Carlo sample paths $\{x_{1:T}^{(i)}\}_{i=1}^{N}$ are already drawn from the posterior distribution. We note that in addition to the algorithm used to estimate $x_{1:T}^{*}$ in (5.1), the quality of the samples and the posterior distribution are also important in improving the accuracy.

Note that when we use the Boltzmann-like target distributions as discussed in Chapter 4, the MLP is the same under different choices of κ, the "temperature" parameter. However, the distribution $\pi(x_{1:T} \mid y_{1:T}, \kappa)$ is more flat for small κ (high temperature) and is more concentrated around the MLP for large κ. Hence the empirical MAP path tends to be more accurate if the Monte Carlo samples are generated from the target distribution with large κ. When κ is sufficiently large, the average sample path is also a good estimate of the MAP. However, it is much more difficult to generate Monte Carlo samples with large κ due to the tendency of being trapped in local optima. The simulated annealing approach provides a natural bridge to link the high temperature system with easily generated samples with the low temperature system with more accurate estimates.

5.2 Annealed SMC

We propose a simulated annealing algorithm for sequential Monte Carlo on state space models. The idea comes from the thermodynamics analogue discussed in the previous section. When the function $\xi(\cdot)$ is chosen to be Boltzmann-like as in (4.2), the Monte Carlo samples from the emulated state space models correspond to a random sample set from the non-interacting particles in a thermodynamic equilibrium system as discussed in Section 4.2.

If the temperature cools down to 0 slowly enough such that the system is approximately in thermodynamic equilibrium for any temperature in between, all particles will condense to the base energy level. The idea of simulated annealing to analogize the physical system was proposed and discussed in Kirkpatrick et al. (1983).

To mimic the thermodynamic procedure, we propose the following system to simulate the annealing procedure for the SMC samples. Let $0 < \kappa_0 < \kappa_1 < \cdots < \kappa_K$ be an increasing

sequence of inverse temperatures. Suppose at κ_0, a base emulated state space model is constructed as

$$\pi(x_{1:T} \mid y_{1:T}; \kappa_0) \propto e^{-\kappa_0 f(x_{1:T})} \propto f_0(x_0) \prod_{t=1}^{T} f_t(x_t \mid x_{1:t-1}) g_t(y_t \mid x_t). \tag{5.4}$$

At a higher inverse temperature κ_k, an emulated state space model can be induced from (5.4) such that

$$\pi(x_{1:T} \mid y_{1:T}; \kappa_k) \propto e^{-\kappa_k f(x_{1:T})} \propto f_0(x_0; \kappa_k) \prod_{t=1}^{T} f_t(x_t \mid x_{1:t-1}; \kappa_k) g_t(y_t \mid x_t; \kappa_k), \tag{5.5}$$

where

$$f_t(x_t \mid x_{1:t-1}; \kappa_k) \propto [f_t(x_t \mid x_{1:t-1})]^{\kappa_k/\kappa_0} \quad \text{and} \quad g_t(y_t \mid x_t; \kappa_k) \propto [g_t(y_t \mid x_t)]^{\kappa_k/\kappa_0}$$

are the corresponding state equations and observation equations at κ_k. The starting inverse temperature κ_0 is usually chosen to be relatively small such that the function $\pi(x_{1:T} \mid y_{1:T}; \kappa_0) \propto e^{-\kappa_0 f(x_{1:T})}$ is relatively flat and is easy to sample from by SMC. We start with κ_0, draw $\{(x_{0,1:T}^{(i)}, w_{0,T}^{(i)})\}_{i=1}^{N}$ from the base emulated state space model $\pi(x_{1:T} \mid y_{1:T}; \kappa_0)$. For $k = 1, \ldots, K$, new samples $\{(x_{k,1:T}^{(j)}, w_{k,T}^{(j)})\}_{j=1}^{M}$ are drawn with respect to the distribution $\pi(x_{1:T} \mid y_{1:T}; \kappa_k)$ utilizing samples $\{(x_{k-1,1:T}^{(j)}, w_{k-1,T}^{(j)})\}_{j=1}^{M}$ obtained at κ_{k-1}. The procedure is depicted in Algorithm 7. The annealed sequential Monte Carlo uses the following proposal distribution at temperature κ_k:

$$q_{k,t}(x_t \mid x_{1:t-1}; \kappa_k) \propto \hat{p}_{k,t}(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1}), \tag{5.6}$$

where the conditional distribution $\hat{p}_{k,t}(x_t \mid x_{1:t-1}; \kappa_{k-1})$ is an estimate of $\pi(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1})$ and can be obtained from the Monte Carlo samples $\{(x_{k-1,1:T}^{(j)}, w_{k-1,T}^{(j)})\}_{j=1}^{M}$ under κ_{k-1}. We will discuss how to obtain such an estimate later. Since κ increases slowly, $\pi(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1})$ and $\pi(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_k)$ are reasonably close.

With a sufficiently large κ_K, samples from the target distribution $\pi(x_{1:T} \mid y_{1:T}; \kappa_K)$ are highly concentrated around the true optimal path $x_{1:T}^*$ and hence are useful in inferring the

Algorithm 7: Annealed Sequential Monte Carlo

Draw $\{(x_{0,1:T}^{(i)}, w_{0,T}^{(i)})\}_{i=1}^{N}$ from $\pi(x_{1:T} \mid y_{1:T}; \kappa_0)$ with SMC in Algorithm 2, using a set of proposal distributions $q_{1,t}(x_t \mid x_{1:t-1}; \kappa_0)$;

for $k = 1, \ldots, K$ **do**

  Draw $\{(x_{k,1:T}^{(j)}, w_{k,T}^{(j)})\}_{j=1}^{M}$ from $\pi(x_{1:T} \mid y_{1:T}; \kappa_k)$ with SMC in Algorithm 2 using the proposal distribution

$$q_{k,t}(x_t \mid x_{1:t-1}; \kappa_k) \propto \hat{p}_{k,t}(x_t \mid x_{k,1:t-1}^{(j)}, y_{1:T}),$$

  where the right hand side is an estimate of $\pi(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1})$;

end

Obtain an estimate of the most likely path from $\{(x_{K,1:T}^{(j)}, w_{K,T}^{(j)})\}_{j=1}^{M}$.

most likely path. However, sampling from $\pi(x_{1:T} \mid y_{1:T}; \kappa_K)$ directly is usually difficult due to the challenge in finding appropriate proposal distributions, which significantly affects the Monte Carlo sample quality. Annealed SMC provides an iterative procedure to the difficult sampling problem under κ_K by utilizing the samples obtained at higher temperature. On one hand, annealed SMC provides a relatively "flat" and easy-sampling starting distribution $\pi(x_{1:T} \mid y_{1:T}; \kappa_0)$ and designs a slow-changing path connecting $\pi(x_{1:T} \mid y_{1:T}; \kappa_0)$ to the desired "sharp" distribution $\pi(x_{1:T} \mid y_{1:T}; \kappa_K)$. On the other hand, for each iteration $k = 1, \ldots, K$, annealed SMC adopts an optimal proposal distribution $p(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1})$, which incorporates the full information set $y_{1:T}$ and is usually difficult to evaluate in conventional SMC implementations. In annealed SMC, the proposal distribution is estimated by sample paths from the previous iteration. The details in estimating the proposal distribution will be discussed in Section 5.3.

The conventional simulated annealing algorithm (Kirkpatrick et al., 1983) is a variation of Markov Chain Monte Carlo (MCMC), which adapts Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) with an extra temperature control. The convergence of the conventional simulated annealing algorithm is given by Granville et al. (1994). However, different from the conventional simulated annealing, annealed SMC does not require for a mixing condition as usually shown in MCMC algorithms. At each iteration at κ_k, the samples are always properly weighted with respect to the target distribution $\pi(x_{1:T} \mid y_{1:T}; \kappa_k)$ because of the weight adjustments. The convergence of SMC samples is

discussed in Crisan and Doucet (2000).

5.3 Practical Issues

In annealed SMC, at temperature $1/\kappa_k$, we need to estimate the proposal distribution $q_{k,t}(x_t \mid x_{0:t-1}; \kappa_k) = \hat{p}_{k,t}(x_t \mid x_{t-1}, y_{1:T})$ with the sample paths from the previous iteration $\{(x^{(j)}_{k-1,T}, w^{(j)}_{k-1,T})\}_{j=1}^M$. Notice that, the weighted samples $\{(x^{(j)}_{k-1,T}, w^{(j)}_{k-1,T})\}_{j=1}^M$ follow the distribution $\pi(x_{1:t} \mid y_{1:T}; \kappa_{k-1})$. Therefore, estimating the proposal distribution is equivalent to estimating the conditional distribution from a sample set drawn from the joint distribution. Here we mention two methods to sample from such a conditional probability.

Parametric Approach. For each time t, suppose $\{\Psi_{t,\theta}(\cdot)\}$ is a parametric family of distributions defined on $\mathcal{X}^t$ and indexed by θ. The joint distribution of $x_{1:t}$ conditioned on $y_{1:T}$ under κ_{k-1} is approximated by one of the distributions in the family. Specifically, let

$$\theta^*_{t,k-1} = \arg\max_{\theta} \prod_{j=1}^{M} w^{(j)}_{k-1,T} \log \psi_{t,\theta}(x^{(j)}_{k-1,1:t}),$$

where $\psi_{t,\theta}$ is the corresponding probability density/mass function of $\Psi_{t,\theta}$. Denote the conditional probability induced from $\Psi_{t,\theta}(x_{1:t})$ as $\psi_{t,\theta}(x_t \mid x_{1:t-1})$. The joint distribution of $x_{1:t} \mid y_{1:T}, \kappa_{k-1}$ is approximated by $\psi_{t,\theta^*_{t,k-1}}(x_{1:t})$ and the proposal distribution $q_{k,t}(x_t \mid x_{1:t-1})$ is estimated by $\psi_{t,\theta^*_{t,k-1}}(x_t \mid x_{1:t-1})$.

One common choice for the distribution family is the multivariate Gaussian distributions. In this case,

$$\psi_{t,\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_{1:t,1:t}}(x_{1:t}) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{1:t,1:t}).$$

The optimal parameter can be obtained by sample mean and sample variance such that

$$\begin{aligned}
\boldsymbol{\mu}^*_{t,k-1} &= \sum_{i=1}^{m} w^{(i)}_{k-1,T} x^{(i)}_{k-1,1:t} \Big/ \sum_{i=1}^{m} w^{(i)}_{k-1,T}, \\
\boldsymbol{\Sigma}^*_{0:t,0:t,k-1} &= \sum_{i=1}^{m} w^{(i)}_{k-1,T} x^{(i)}_{k-1,1:t} \left[x^{(i)}_{k-1,1:t}\right]' \Big/ \sum_{i=1}^{m} w^{(i)}_{k-1,T}.
\end{aligned}$$

Denote

$$\boldsymbol{\mu}^*_{t,k-1} = \begin{pmatrix} \boldsymbol{\mu}^*_{t-1,k-1} \\ \mu^*_{t,k-1} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}^*_{1:t,1:t,k-1} = \begin{bmatrix} \boldsymbol{\Sigma}^*_{1:t-1,1:t-1,k-1} & \boldsymbol{\Sigma}^*_{1:t-1,t,k-1} \\ \boldsymbol{\Sigma}^*_{t,1:t-1,k-1} & \Sigma^*_{t,t,k-1} \end{bmatrix}.$$

Then the induced conditional probability has the following closed-form:

$$p(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1}) = \mathcal{N}\left(\mu_{t|1:t-1,k-1}, \Sigma_{t|1:t-1,k-1}\right),$$

where the parameters are

$$\begin{aligned} \mu_{t|1:t-1,k-1} &= \mu^*_{t,k-1} + \boldsymbol{\Sigma}^*_{t,1:t-1,k-1}\left[\boldsymbol{\Sigma}^*_{1:t-1,1:t-1,k-1}\right]^{-1}(x_{1:t-1} - \boldsymbol{\mu}^*_{t-1,k-1}), \\ \Sigma_{t|1:t-1,k-1} &= \Sigma^*_{t,t,k-1} - \boldsymbol{\Sigma}^*_{t,1:t-1,k-1}\left[\boldsymbol{\Sigma}^*_{1:t-1,1:t-1,k-1}\right]^{-1}\boldsymbol{\Sigma}^*_{1:t-1,t,k-1}. \end{aligned}$$

The results above for multivariate Gaussian distributions can be easily extended to mixture Gaussian distributions, which can approximate most distributions well.

Nonparametric Approach. When there is no appropriate distribution family to describe the joint distribution of $x_{k-1,1:t} \mid y_{1:T}$, one can sample from the conditional distribution $p(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1})$ of $\{(x^{(j)}_{k-1,1:T}, w^{(j)}_{k-1,T})\}_{j=1}^M$ nonparametrically. Specifically, suppose $\mathcal{K}_{\boldsymbol{b}_1}(\cdot)$ and $\mathcal{K}_{b_2}(\cdot)$ are kernel functions for $x_{1:t-1}$ and x_t, respectively, and it is easy to sample from $\mathcal{K}_{b_2}(\cdot)$. For any given $x^{(j)}_{k,t-1}$, Algorithm 8 depicts the nonparametric approach to draw $x^{(j)}_{k,t}$ from the conditional distribution $p(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1})$ when the samples $\{(x^{(j)}_{k-1,1:T}, w^{(j)}_{k-1,T})\}_{i=1}^M$ properly weighted to $\pi(x_{1:T} \mid y_{1:T}; \kappa_{k-1})$ are available.

Algorithm 8: Sample nonparametrically from a Empirical Conditional Distribution

for *any* $x^{(j)}_{k,1:t-1}$ **do**
  Draw l from $\{1, \ldots, M\}$ with probabilities proportional to

$$P(l = i) \propto w^{(i)}_{k-1,T}\mathcal{K}_{\boldsymbol{b}_1}(x^{(i)}_{k-1,1:t-1} - x^{(j)}_{k,1:t-1})$$

  Draw ε from the density induced by $\mathcal{K}_{b_2}(\cdot)$;
  return $x^{(j)}_{k,t} = x^{(l)}_{k-1,t} + \varepsilon$.
end

The parametric approach often requires the state space model to satisfy certain conditions. For example, when both state equations and observation equations are approximately linear and Gaussian, the multivariate Gaussian distribution family can be used to estimate the conditional distributions. The nonparametric approach can deal with general state space models. However, it often costs much more computing power than the parametric approach.

One issue for both approaches is the high dimensionality. Unless the system has a short memory, the conditional distribution at time t involves the high dimensional $x_{1:t}$ and with potentially increasing dimension of parameters needed or the the dimensions of spaces the nonparametric approach need to operate within. One solution for reducing the dimension of the sampling problem is to use a low-dimensional sufficient statistics. Suppose $S(x_{1:t-1})$ is a low-dimensional sufficient statistic such that $p(x_t \mid x_{1:t-1}, y_{1:T}; \kappa_{k-1}) = p(x_t \mid S(x_{1:t-1}), y_{1:T}; \kappa_{k-1})$. Both parametric and nonparametric approaches can therefore be conducted on the joint distribution of $(x_t, S(x_{1:t-1}))$, which is of lower dimension. In a Markovian system, $S(x_{1:t-1}) = x_{t-1}$ and the problem reduces to sampling from a much simpler distribution. In an auto-regressive system with lag δ, $S(x_{1:t-1}) = x_{t-\delta:t-1}$, which is a $\delta + 1$-dimensional system. Note that since the estimated conditional distribution is used as a proposal distribution, it is often tolerable to use less accurate estimators for computational efficiency. Hence various approximation and dimension reduction tools can be used, including variational Bayes approximations (Tzikas et al., 2008).

Another issue in estimating the conditional distribution from sequential Monte Carlo samples is the sample degeneracy. In SMC, degeneracy refers to the phenomenon that the number of distinct values for some states such as X_1 can be less than the number of Monte Carlo samples, if resampling steps are engaged. The degeneracy problem is crucial for both approaches in sampling from the conditional distribution. Therefore, at $\kappa > \kappa_0$, we suggest to conduct resampling only when all propagation steps are finished to prevent the samples from trapping into local maximums. When high degeneracy is persistent, we suggest to use post-MCMC steps (Gilks and Berzuini, 2001) to regenerate the samples. If the system is reversible and SMC can be implemented backward in t, alternating forward and backward sampling through the annealing iterations may also reduce the degeneracy problem as it

starts with more diversified samples in each temperature iteration.

5.4 Path refinement with Viterbi algorithm

A more accurate estimate of the mode can be obtained by using Viterbi algorithm (Viterbi, 1967) on the discrete space consisting of the SMC samples. The Viterbi algorithm is a dynamic programming algorithm originally used to solve the MLP problem in hidden Markov models, where the hidden states are finite. Let $\mathcal{A}_t = \{a_t^{(j)}\}_{j=1}^M$ be the grid points for x_t and $\Omega = \mathcal{A}_1 \times \cdots \times \mathcal{A}_T$ be the Cartesian product of the grid point sets. In state space models, the Viterbi algorithm searches for the maximum over all possible combinations of the grid points in Ω. Specifically, the MLP obtained by the Viterbi algorithm is

$$\hat{x_{1:T}}^{(viterbi)} = \arg\max_{x_{1:T} \in \Omega} \pi(x_{1:T} \mid y_{1:T}). \tag{5.7}$$

The Viterbi algorithm for state space models based on the grid points $\{a_1^{(j)}\}_{j=1}^M, \ldots, \{a_T^{(j)}\}_{j=1}^M$ is depicted in Algorithm 9.

The SMC samples drawn from the emulated state space model provide a set of grid points for the Viterbi algorithm. For example, one can set $\mathcal{A}_t = \{x_t^{(j)}\}_{j=1}^M$ such that $\Omega = \{x_1^{(j)}\}_{j=1}^M \times \cdots \times \{x_T^{(j)}\}_{j=1}^M$ is the joint set of all SMC sample points. One can also add and remove grids points to expand coverage with more details around the more important state paths.

The Viterbi algorithm explores all combinations of sample points and results in a better mode estimation compared with the empirical MAP in (5.2). However, it has its limitations for implementation with state space models. One limitation is that the Viterbi algorithm only works on Markovian state space models. In addition, it only works with a non-singular state evolution in which the degree of freedom is the same as the state variable dimension. Otherwise, state paths cannot be re-assembled as the Viterbi algorithm tries to achieve. For example, in the cubic spline problem, the state evolution is singular. Although one can reduce the dimension of the state variable to make the evolution non-singular, the state evolution then becomes non-Markovian. Another limitation is the requirement for Monte

Algorithm 9: Viterbi Algorithm for Markovian State Space Models

Let $\mathcal{A}_t = \{a_t^{(j)}\}_{j=1}^M$ be a set of grid points for x_t for $t = 1, \ldots, T$;

Initialize $\ell_0^{(j)} = 0$ and $\hat{x}_1^{(j)} = a_1^{(j)}$ for $j = 1, \ldots, M$;

for $t = 2, \ldots, T$ **do**

  for $j = 1, \ldots, M$ **do**

    Set

$$\ell_t^{(j)} \leftarrow \max_{k \in \{1,\ldots,M\}} \ell_{t-1}^{(k)} f_t(a_t^{(j)} \mid \hat{x}_{1:t-1}^{(k)}) g_t(y_t \mid a_t^{(j)}) \tag{5.8}$$

    Set

$$\hat{x}_{1:t}^{(j)} = (\hat{x}_{1:t-1}^{(k_j^*)}, a_t^{(j)}),$$

    where k_j^* is the optimal point of (5.8).

  end

end

Let

$$j^* = \arg\max_{j \in \{1,\ldots,M\}} \ell_T^{(j)}.$$

return $\hat{x}_{1:T}^{(j^*)}$

Carlo sample size. The Monte Carlo samples induced Ω provide a discretization of the support $\mathcal{X}$ for each time t. The accuracy of the Viterbi algorithm strongly dependents on the discretization quality, especially when $\mathcal{X}$ is continuous. In general, the denser the Monte Carlo samples are around the true MLP, the more accurate the Viterbi algorithm solution is. As a result, it often requires a large Monte Carlo sample size to generate better discretization and to achieve high accuracy with the Viterbi algorithm. To reduce the path error $\|\hat{x}_{1:T}^{(viterbi)} - x_{1:T}^*\|$ by half, the Monte Carlo sample size M needs to be doubled, because the discretization size is reduced by half on average with double sample size. On the other hand, the computational cost increases quadratically with the sample size M. One possible way to improve is to apply the Viterbi algorithm iteratively by shrinking to the high value region of last iteration and regenerating grid points there. Similar to iterative grid search, the iterative Viterbi algorithm may result in a sub-optimal solution.

5.5 Simulation Examples

5.5.1 Cubic Smoothing Spline

In this simulation study, we consider the cubic smoothing spline problem in Section 4.3.1. The observations are generated by

$$y_t = \sin(9(t-1)/100) + \zeta_t,$$

for $t = 1, \ldots, 50$, with $\zeta_t \sim \mathcal{N}(0, 1/16)$ and we fix $\lambda = 10$ in the objective function (4.3).

Since the dynamic system is linear and Gaussian, the most likely path is obtained by Kalman Smoother (Kalman, 1960). We use it as the benchmark. We start from the initial inverse temperature $\kappa = \kappa_0 = 4$. Figure 5.1 demonstrates $m = 1000$ samples (in grey) drawn from the target distribution $\pi(x_T \mid y_{1:T}; \kappa_0) \propto [\pi(x_{1:T} \mid y_{1:T})]^{\kappa_0}$ by the SMC algorithm described in Algorithm 2 along with the observations $y_{1:T}$ (the solid line) and the true most likely path (the dashed line).

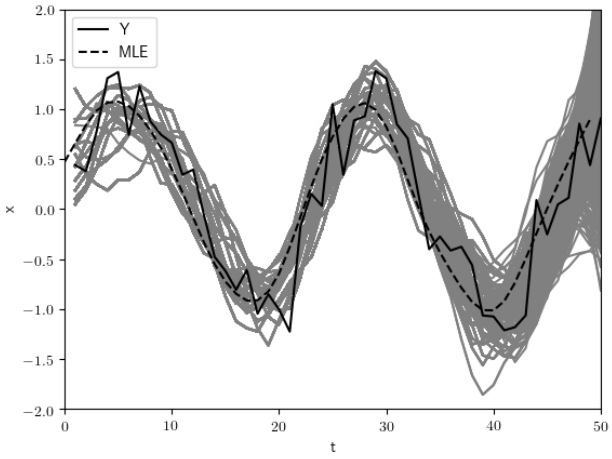


Figure 5.1: Sample paths at $\kappa_0 = 4$.

The proposal distribution $q_t(\cdot)$ used at κ_0 is chosen to be proportional to $f_t(x_t \mid x_{1:t-1})$ $g_t(y_t \mid x_t)$. At each time t, η_t is drawn from the proposal distribution $q_t(\eta_t \mid a_{t-1}, b_{t-1}, c_{t-1}, y_t)$, which is a Gaussian distribution in this case. Resampling is conducted when the effective sample size defined in (2.11) is less than $0.3m$.

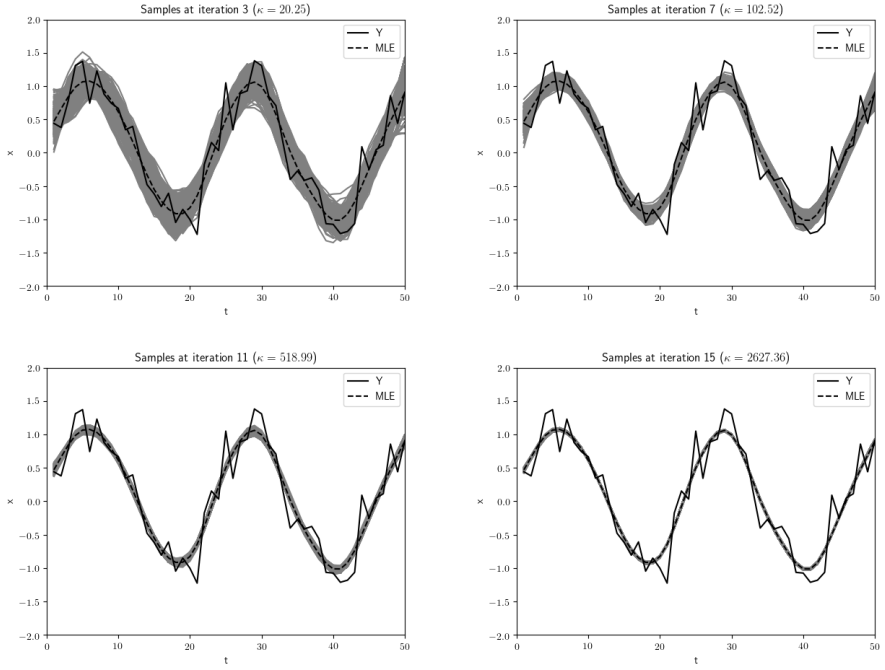


Figure 5.2: Sample paths at different κ's

To find the most likely path stochastically and numerically, we apply the annealed SMC approach in Algorithm 7 with a predetermined sequence of inverted temperatures $\kappa_k = 1.5^k \kappa_0$ for $k = 1, \ldots, 16$. The proposal distribution for the anneal SMC is estimated by the parametric approach. Specifically, since the innovation in the state equation is of one dimension, at κ_k, we only need to generate proposal samples for c_t. It is drawn by first fitting $\{(c_{k-1,t}^{(j)}, a_{k-1,t-1}^{(j)}, b_{k-1,t-1}^{(j)}, c_{k-1,t-1}^{(j)})\}_{j=1}^{m}$ with a multivariate Gaussian distribution and then sampling from the conditional distribution. To prevent degeneracy, resampling step is only conducted at the end of each annealing SMC iteration and after each iteration, one step of post-MCMC move is conducted to regenerate sample states. The post-MCMC move uses blocked Gibbs sampling (Jensen et al., 1995), due to the special structure of the state dynamic. At each iteration of the Gibbs sampling, (x_t, x_{t+1}, x_{t+2}) are updated together.

Figure 5.2 shows the sample paths (after the post-MCMC step) at the end of different anneal SMC iterations. When the temperature shrinks to zero as κ increases, the sample paths move to a small neighbor region around the true most likely path. Figure 5.3 shows the value of the objective function at the weighted average path of the samples as for different numbers of iterations. The true optimal value (the objective function value at the optimal

path) obtained by the Kalman smoother is plotted as the dashed horizontal line. As the number of iteration increases, the objective function value at the averaged path decreases stochastically and convergences at roughly the 7th iteration.

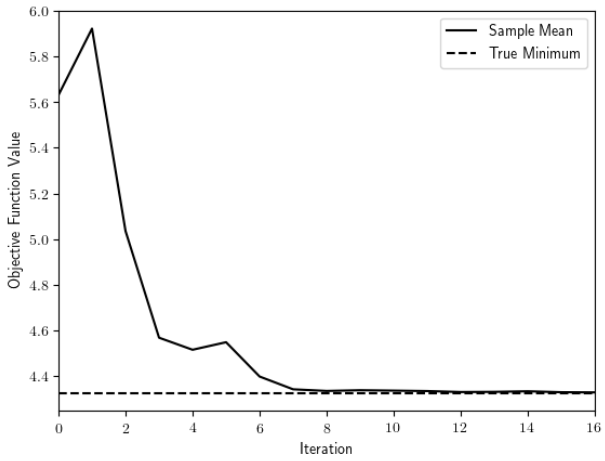


Figure 5.3: Value of the objective function against the number of iterations

To compare the computational efficiency, we record the computing time needed for different approaches, shown in Table 5.1. The Scipy approach uses the nonlinear optimizer provided by the python package Scipy (Jones et al., 2001), which implements the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm by default. The annealed SMC records the time until convergence (the time when the value of the objective function is not improved by further iteration). Kalman Smoother is the fastest one due to its deterministic nature in finding the most likely path for linear Gaussian models. Annealed SMC is slower than the nonlinear solver program provided by Scipy, but achieves similar accuracy. We also note that this is a simple convex optimization problem in which a straightforward optimization algorithm such as the Scipy performs well. Our estimation approach is more flexible and this example serves as an illustration of how the algorithm works.

Kalman Smoother	Scipy minimizer	Annealed SMC
2.2 ms	129.6 ms	232.9 ms

Table 5.1: Time spent by different approaches.

5.5.2 LASSO Regression

In this simulation study, we consider the LASSO regression problem as discussed in Section 4.3.2. We set $n = 40$ observations, $p = 20$ covariates and $\sigma_y = 0.3$. The covariates $(Z_1, \ldots, Z_p)$ are generated from a multivariate normal distribution $\mathcal{N}(0, \Sigma)$ where all diagonal elements of Σ is 1 and all off-diagonal elements are 0.4. β's are generated i.i.d. according to Bernoulli(0.2). λ is set to 5 in the objective function (4.10).

We start from the initial emulated model with the temperature parameter $\kappa = \kappa_0 = 0.05$. $m = 5000$ samples are drawn from the standard SMC algorithm under the target distribution (4.13) with $\kappa_0 = 0.05$. The state equation (4.14) is used as the proposal distribution and the weight is from the observation equation (4.15) as a consequence. Resampling is done when the effective sample size in (2.11) is below $0.3m$. The sampled state paths are plotted in Figure 5.4. The estimated path for solving the original LASSO problem (4.10) using the scikit-learn python package (Pedregosa et al., 2011) is treated as the benchmark.

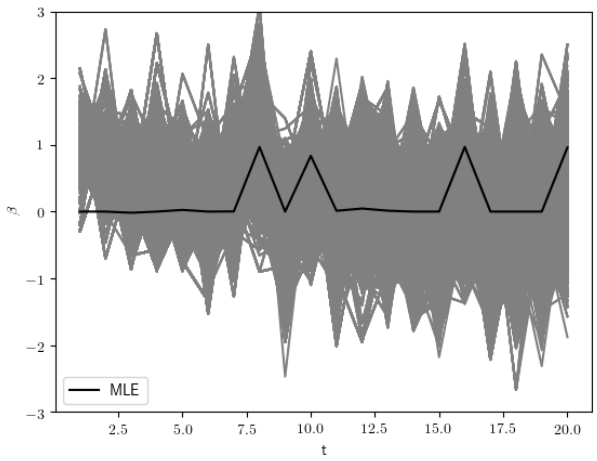


Figure 5.4: Sample paths at $\kappa_0 = 0.05$

In the subsequent annealing procedure, we use $m = 2000$ samples and set $\kappa_k = 1.5^k \kappa_0$ for $k = 1, \ldots, 30$. The proposal distribution used in the annealing procedure is estimated with a multivariate normal approximation of the joint distribution of $(\beta_{k-1,t}, \ldots, \ldots, \beta_{k-1,1})$. Resampling is done only at the end of each iteration and 10 steps of post-MCMC runs are applied. The post-MCMC runs use the Gibbs sampling approach with the Metropolis-

Hasting transition kernel (Metropolis et al., 1953; Hastings, 1970), where for $t = 1, \ldots, T$ and for $i = 1, \ldots, m$, a new value for β_t is proposed such that $\tilde{\beta}_t^{(i)} = \beta_t^{(i)} + \mathcal{N}(0, \tau^2)$, where $\tau^2 \propto 1/\kappa$, and the proposed move is accepted with the probability $\min\{1, \pi(\tilde{x}_{1:t}^{(i)} \mid y_{1:T}; \kappa)/\pi(x_{1:T}^{(i)} \mid y_{1:T}; \kappa)\}$ with $\tilde{x}_{1:t}^{(i)} = (x_{1:t-1}^{(i)}, \tilde{x}_t^{(i)}, x_{t+1}^{(i)}, \ldots, x_T^{(i)})$. Figure 5.5 plots the sample paths at four different levels of κ's. Again, it is seen that the procedure is able to gradually move the sample paths towards the optimal solution. Figure 5.6 shows the convergence of the values of the objective function in (4.10) evaluated at the weighted average of the sample paths.

After around 17 iterations, the weighted mean of the samples generated from the annealed SMC converges. Due to Monte Carlo variations, the sample paths and the average path cannot shrink the coefficients to exactly zero. It is tempting to run the Viterbi algorithm to refine the estimate, with zeros added to the set of allowed values of the state variables. Unfortunately the state space model designed for the LASSO problem is not Markovian hence the Viterbi algorithm cannot be used. However, we used an additional refinement step by iteratively and greedily comparing each estimated state $\hat{x}_t$ (using the average sample path) with zero under the original objective function. The refinement step (with additional 0.063ms in computing time) moved some of the states to zero, and improved the value of the objective function from 21.90356 to 21.899657. The minimum achieved by the Scikit solver is 21.899645. However, such a refinement is based on the knowledge that the solution of Lasso has exactly zero coefficients, and may not be used in other optimization problems. Note that, the emulation system can be easily generalized to other types of regularization on parameters by changing the penalty term in (4.15) without much effort and can be adapted much more complex penalty structure.

5.5.3 Optimal Trading Path

In this simulation, we consider the optimal trading path problem in Section 4.3.3. Similarly, we set $T = 20$, $\sigma_x^2 = 0.25$, $\sigma_y^2 = 1$ and $\alpha = 0.5$. The ideal trading path is given by

$$y_t = 25\exp\{-(t+1)/8\} - 40\exp\{-(t+1)/4\}.$$

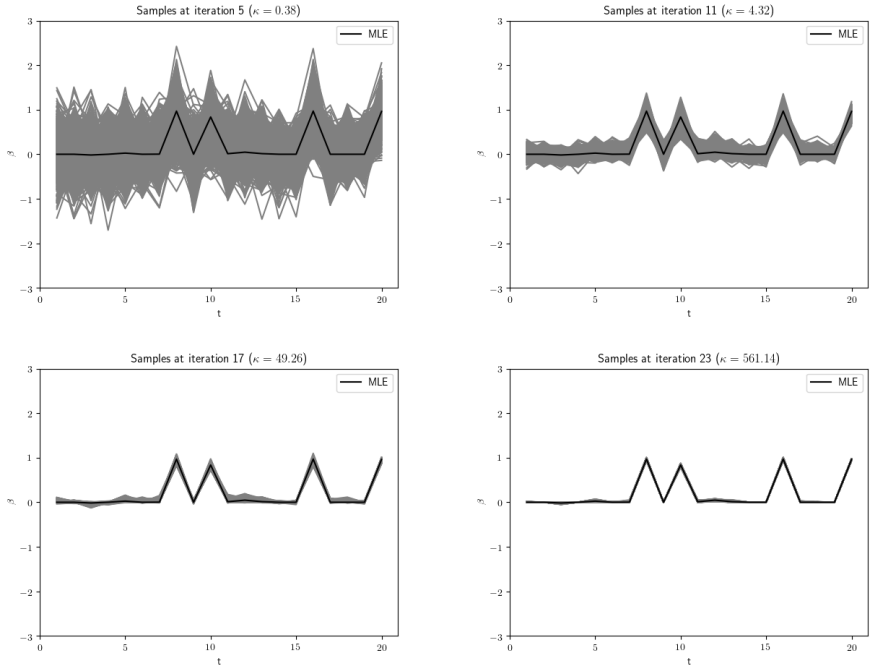


Figure 5.5: Sample paths at different κ's

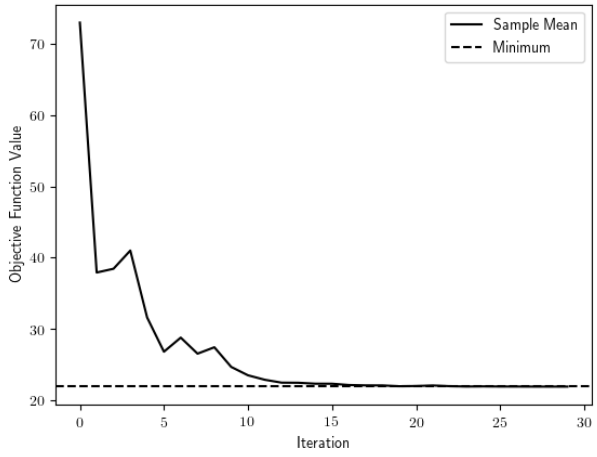


Figure 5.6: Value of the objective function against the number of iterations

We start from the initial temperature $\kappa = \kappa_0 = 1.0$. The sample paths at κ_0 is drawn with the constrained SMC in Algorithm 3 as in the example of Section 3.6. In this example, we use $m^* = 300$ backward pilot samples. The resulting $m = 1000$ (forward) sample paths are shown in Figure 5.7. The observations $y_1, \ldots, y_T$, which represent the ideal optimal trading strategy without the trading cost, are plotted as the solid line. An estimated path, marked by a dashed line, is provided by the Scipy nonlinear optimization algorithm.

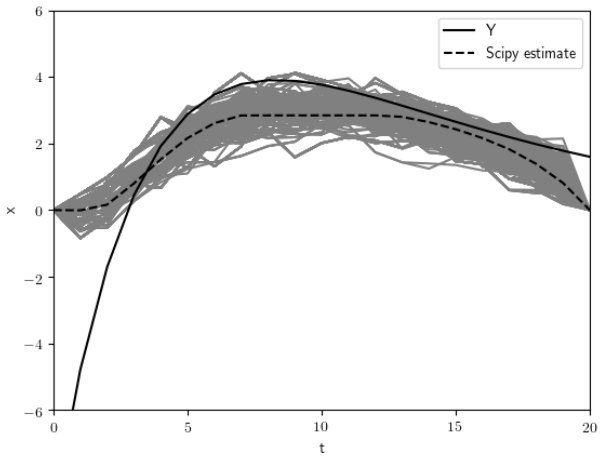


Figure 5.7: Sample paths at κ_0

We use the following sequence of inverted temperatures for annealing: $\kappa_k = 2^k \kappa_0$ for $k = 1, \ldots, 20$. The proposal distribution in the annealed SMC is sampled with the parametric approach by approximating the joint distribution of $x_{k-1,t}$ and $x_{k-1,t-1}$ with a bivariate normal distribution. The annealed $m = 1000$ sample paths are resampled at the end of each iteration, and no post-MCMC step is conducted. Samples at several different inverted temperatures are shown in Figure 5.8. We use the sample average as our estimator for the most likely path. The value of the objective function at the sample average path decreases stochastically as shown in Figure 5.9. It eventually converges at around the 11th iteration. The optimal objective function value achieved by the annealed SMC is 89.459, while the one obtained by the Scipy nonlinear optimizer is 89.462. The values of the objective function at the sample paths at the 20th iteration have an average of 89.459 and a standard deviation of 1.09×10^{-5}. The annealed SMC gains some improvement in accuracy at the cost of extra computation. The Scipy nonlinear optimizer takes 78ms while the annealed SMC costs

1.820 seconds for the initial emulated model (including the time of backward sampling) and costs around 2ms for each subsequent iteration. Sampling from the base emulated model costs much more than subsequent iteration for two reasons. First, it requires a large sample size for the base model because of high degeneracy. Second, the end point constraint is imposed and an additional backward pilot run is needed to reduce degeneracy.

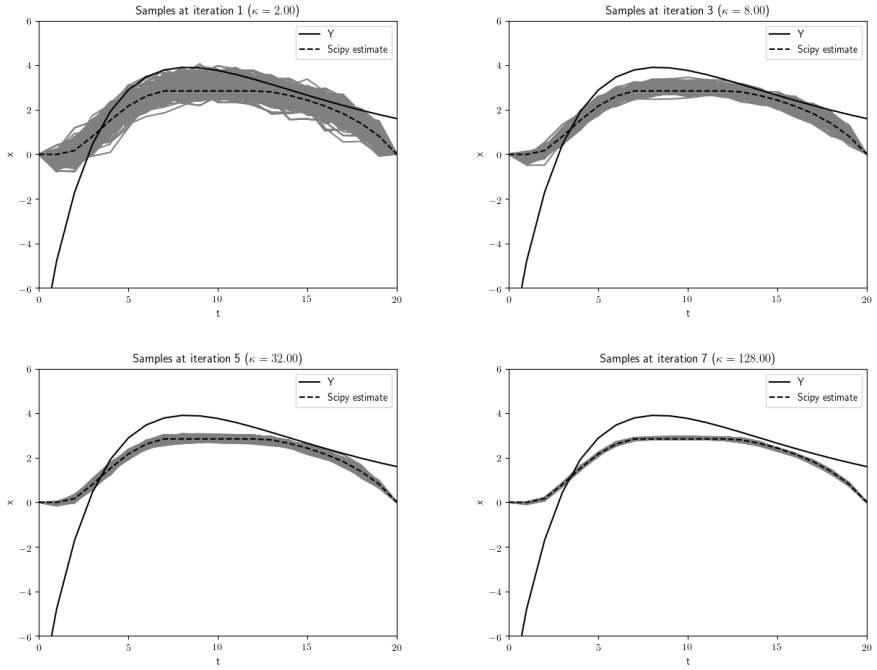


Figure 5.8: Sample paths at different κ's

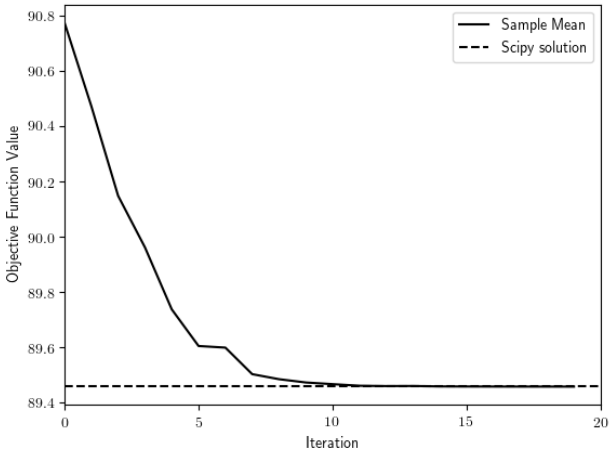


Figure 5.9: Value of the objective function against the number of iterations

5.5.4 L1 Trend Filtering

In this simulation study, we consider the ℓ_1 trend filtering problem in Section 4.3.4. We set $T = 60$, $\lambda = 10$ and

$$y_t = \begin{cases} \dfrac{t-1}{20} + \mathcal{N}(0, 0.01), & 1 \leqslant t \leqslant 20 \\ \dfrac{40-t}{20} + \mathcal{N}(0, 0.01), & 21 \leqslant t \leqslant 40 \\ \dfrac{t-41}{20} + \mathcal{N}(0, 0.01), & 41 \leqslant t \leqslant 60. \end{cases}$$

At $\kappa = \kappa_0 = 10$, $m = 5000$ SMC paths are sampled using the state dynamics (4.22) as the proposal distribution. A resampling step is conducted when the effective sample size drops below $0.1m$. The approximate MLE marked as dashed line is the solution obtained by Scipy nonlinear solver. The solution shows a piece-wise linear behavior as the $\ell 1$ type of penalty appears in the objective function.

We use the following designed annealing sequence $\kappa_k = 1.3^k \kappa_0$ for $k = 1, \ldots, 40$ and use $m = 2000$ samples for annealing. In each annealing iteration, the proposal distribution used is $Laplace(\hat{E}[x_t \mid x_{t-1}, x_{t-2}; \kappa_k], \hat{V}[x_t \mid x_{t-1}, x_{t-2}; \kappa_k]^{1/2}/\sqrt{2})$ where $\hat{E}$ and $\hat{V}$ are estimated from the samples from the last iteration $\{(x^{(j)}_{k-1,t}, x^{(j)}_{k-1,t-1}, x^{(j)}_{k-1,t-2})\}_{j=1}^m$.

The Laplace distribution has a heavier tail than the normal distribution with the same variance. We found it more efficient to sample from the Laplace distribution to reduce sample degeneracy in this problem. The resampling step is conducted at the end of each iteration and is followed by 10 steps of post-MCMC moves. The post-MCMC steps follow the standard Gibbs sampling as in the LASSO example. Sample paths at four different κ's are displayed in Figure 5.10. Note that when $\kappa \approx 1462$, the sample paths are different from the nonlinear solver's solution at $t \in [38, 42]$. The value of the objective function at the sample average path shown in Figure 5.11 show that annealed SMC can obtain a smaller objective function value than the Scipy optimizer. The Scipy nonlinear optimizer takes 155ms while annealed SMC costs 22 ms for SMC sampling from the initial emulated model and costs around 160 ms for each subsequent annealing iteration including the post-MCMC runs.

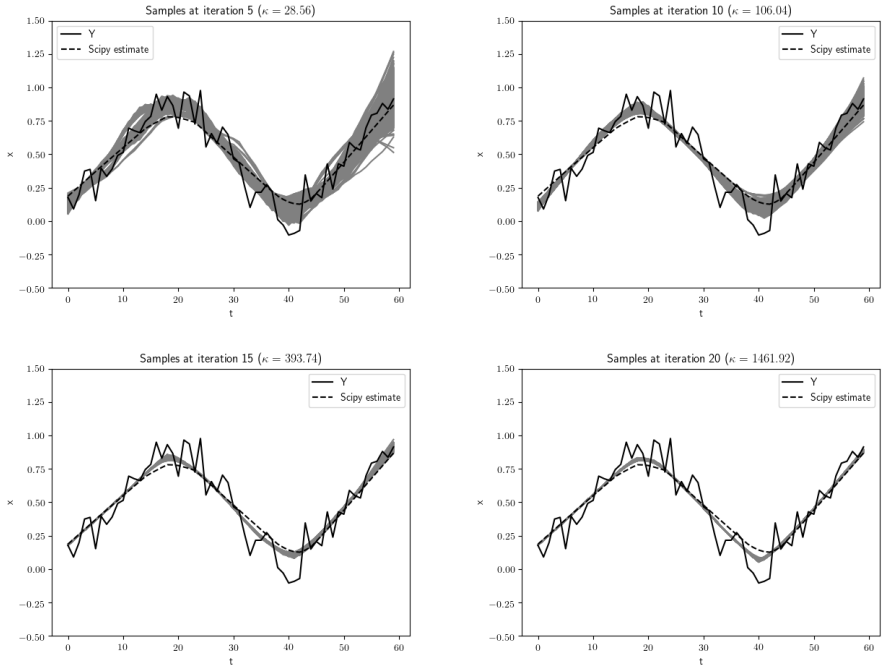


Figure 5.10: Sample paths at different κ's

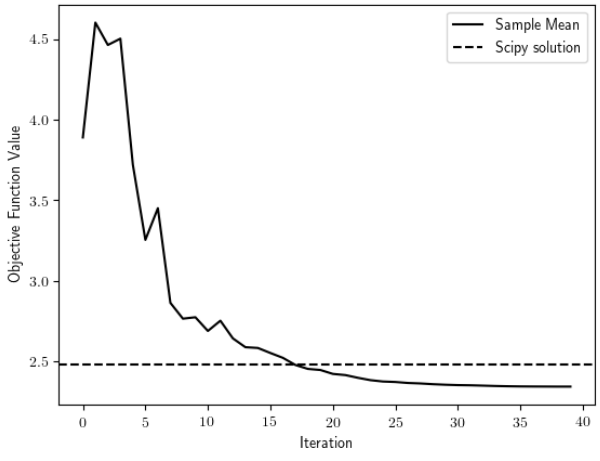


Figure 5.11: Value of the objective function against the number of iterations

PART II

Individualized Group Learning

CHAPTER 6

Introduction to Individualized Inference through Fusion Learning

6.1 Fusion Learning and Individualized Inference

6.1.1 Fusion Learning

Fusion learning is a methodology that aggregates information/dataset from different sources to make a coherent overall inference (Chen and Xie, 2014; Liu et al., 2014, 2015; Yang et al., 2016). It has an old name "meta-analysis", which was named by Glass (1976) as the "analysis of analyses". The main idea is to combine outcomes from different studies on the same object to provide a more powerful inference result.

Based on the idea of combination of information, two nature questions arise: what to combine and how to combine. The p-value based meta-analysis approaches are discussed by Marden (1991), where point summary information (p-value in this case) from different studies of a same object are combined with equal weights. Normand (1999) generalized the p-value based meta-analysis to the class of model-based meta-analysis approaches, including fixed-effects models and random-effects model. The potential unobserved heterogeneity in different studies is properly tackled in the model-based meta-analysis approaches. In fixed-effect models, the parameters of interest are assumed to be unknown and fixed, while those in random-effects models are supposed to be generated randomly from a (super-) population of parameters. Apart from p-value based models of Marden (1991), the summary information of different studies are averaged with unequal weights in model-based

meta-analysis approaches. The weights usually relate to their precision. The varies meta-analysis approaches are unified by Xie et al. (2011) under a framework using the concept of confidence distribution (CD), which contains confidence interval information of all levels for a parameter of interest (Xie et al., 2013; Schweder and Hjort, 2016).

The idea of data fusion or data aggregation is general and has been widely accepted. For example, in statistical learning, the class of ensemble learning approaches fits each model with a random subset of data and make prediction by averaging the output of those models with equal or unequal weights to improve accuracy (Opitz and Maclin, 1999; Polikar, 2006).

6.1.2 Individualized Inference

The individualized inference problem originates from researches in precision medicine, whose goal is to provide an optimal treatment suggestion tailored to the particular situation of one specified patient (Insel, 2009; Hamburg and Collins, 2010). It is challenging to find other patients that are exactly identical to the target patient to carry out statistical analysis on the treatment effects. As a compromise, the optimality of a treatment can be only measured over the average effect of a group of similar patients (Qian and Murphy, 2011). The group of similar patients, as the subject of the inference problem, is constructed for the particular patient of interest and is therefore different when the patient of interest changes. The idea of scaling down to a local subset of data neighboring to the target individual also gives rise to the study of target maximum likelihood estimate method proposed by van der Laan and Rubin (2006). Van der Laan and Rose (2011) applies the targeted maximum likelihood approach to causal inference and precision medicine problems to reduce estimation bias.

Analogue to doctor assigning a personalized treatment to the target patient, Liu and Meng (2016) proposed the concept of individualized inference, where an individualized estimate for a target dataset/individual is constructed utilizing information from others. As a comparison, the precision medicine focuses on finding the optimal personalized treatment assignment function in the context of causal inference, while individualized inference is a broader topic containing other inference problems.

6.2 Fusion Learning through Aggregation

In this section, the aggregation-based fusion learning methods will be reviewed. Suppose a sample dataset contains K independent individuals. Each *individual* can be one study, an individual patient, or in some cases a subset of observations according to a partition of the whole sample set, depending on the context of application. Let $\mathcal{S}_k$ be the full information set for individual k. $\mathcal{S}_k$ can be, for example, all the observations from individual k. In addition, assume θ_k is the corresponding parameter of interest for individual k.

In classical fusion learning, it is commonly assumed that all individuals share a common parameter of interest such that $\theta_1 = \theta_2 = \cdots = \theta_K = \theta$. The assumption is often imposed when the individuals are independent studies on the same object or the information aggregation are from random subsets of a single dataset. Normand (1999) relaxed this assumption to the case when $\theta_k, k = 1, \ldots, K$, are i.i.d. realizations of a common distribution with parameter θ of interest.

In this section, we limit our discussion to the classical fusion learning with the identical parameter assumption.

A typical classical fusion learning approach has three steps. In the first step, information regarding the parameter θ_k is summarized from its information set $\mathcal{S}_k$ through an estimating function $m(\theta; \mathcal{S}_k)$. In the second step, a population level aggregated estimating function is calculated by a weighted average of individual estimating functions such that

$$m^{(c)}(\theta) = \frac{\sum_{k=1}^{K} w_k m(\theta; \mathcal{S}_k)}{\sum_{k=1}^{K} w_k}, \tag{6.1}$$

where $w_k \geqslant 0$ are generic weights for the aggregation. In the last step, a point estimator for θ is further inferred from the aggregated estimating function in (6.1).

The weight w_k in (6.1) can be either fixed or dependent on the data $\mathcal{S}_k$, for example, w_k can be set to be proportional to the precision matrix of the (individual level) parameter estimator of individual k. Specifically, when all resources are symmetric in terms of methodology, the number of observations, etc., the weights are set to be equal. When in fixed effects models and random effects models, the optimal choice of weights is proportional

to the precision matrix of each individual.

The estimating function $m(\theta; \mathcal{S}_k)$ is generic concept, which can be log-likelihood function, pseudo-likelihood function, loss function, log confidence distribution function or even the individual level point estimator $\hat{\theta}_k$.

A general framework for combining independent confidence distributions using any given coordinate-wise monotonic function was proposed by Singh et al. (2005). Specifically, Xie et al. (2011); Yang et al. (2016) proposed to use the aggregation form in (6.1) with the estimating function

$$m(\theta; \mathcal{S}_k) = F_0^{-1}(H_k(\theta)), \tag{6.2}$$

where $F_0(\cdot)$ is a given cumulative distribution function and $H_k(\cdot)$ is a confidence distribution for θ_k induced from $\mathcal{S}_k$. Sometimes, the combination of (6.1) and (6.2) can be simplified to

$$m^{(c)}(\theta) = \sum_{k=1}^{K} w_k H_k(\theta), \tag{6.3}$$

when F_0 is the c.d.f. of uniform distribution and w_k's are normalized (Xie et al., 2011).

The confidence distribution based fusion learning framework using (6.3) is generic and powerful with a borad range of applications in challenging problem settings. Examples include robust fusion learning (Xie et al., 2011), discrete data (Liu et al., 2014), heterogeneous individuals (Liu et al., 2015) and split-conquer-combine approaches (Chen and Xie, 2014). Cheng et al. (2017) provides a detailed review on the confidence distribution based fusion learning approaches.

Beyond aggregating confidence distributions, Gao and Carroll (2017) proposed a fusion scheme with pseudo-likelihood functions, where the integrated pseudo-likelihood function is an aggregation of individual pseudo-likelihoods.

$$\ell^{(c)}(\theta) = \sum_{k=1}^{K} w_k \ell_k(\theta_k; \mathcal{S}_k),$$

where $\ell_k(\theta_k; \mathcal{S}_k)$ is the pseudo-likelihood function of individual k. Varin and Vidoni (2006); Joe and Lee (2009) provide data structure based practical strategies for choosing the corresponding weight w_k.

In opposite to the identical assumption, when the population is heterogeneous such that $\theta_1 \neq \theta_2 \neq \cdots \neq \theta_K$ and when estimating θ_1 attracts more interest than estimating the population average θ, the aggregation (6.1) should be modified in order to meet the need of individualized inference. This class of fusion learning-based individualized inference will be discussed in Section 6.3.

6.3 Fusion Learning Based Individualized Inference

When potential heterogeneity exists in the population, the identical parameter assumption in fusion learning often does not hold. Instead of assuming a common parameter of interest for all individuals, individualized inference through fusion learning focuses on improving the inference efficiency of one specific study or individual by borrowing strength from similar studies or individuals. Specifically, suppose the individual of interest is marked as individual 0 with the parameter of interest θ_0, the goal of fusion learning based individualized inference is to provide a better point estimator, through aggregation, than that obtained based only on the data of individual 0.

The main challenge is that bias arises when fusing a heterogeneous population. On one hand, aggregating too many other individuals brings extra bias due to heterogeneity. On the other hand, if few other individuals are fused, variance reduction is limited. Individual level study yielding an estimator, say $\hat{\theta}_0$, is the extreme case with no bias but also no variance reduction. The population-wise fusion learning, where all individuals are fused together with equal weights, is another extreme that maximizes variance reduction but may potentially have a large bias. Individualized version of fusion learning alleviates the problem by taking control over the aggregation weight w_k through a similarity measure between $\mathcal{S}_k$ and $\mathcal{S}_0$. Particularly, the aggregation formula for individualized inference through fusion learning is

$$m_0^{(c)}(\theta) = \frac{\sum_{k=0}^{K} w_{0,k} m(\theta; \mathcal{S}_k)}{\sum_{k=0}^{K} w_{0,k}}, \tag{6.4}$$

where the extra subscript 0 indicates the target individual of such an aggregation. In fusion learning, the aggregation in (6.1) is constructed once and yields one point estimate for all individuals since they share the common parameter of interest. However, in individual-

ized inference, the corresponding individualized aggregation (6.4) is constructed for each individual of interest.

The *i*Fusion approach proposed by Shen et al. (2019) considers the asymptotic settings when the effective sample size for each individual n_k increase to infinite but the proportion converges to some value between 0 and 1 such that $n_k/\sum_k n_k = O_p(1)$. Note that the effective sample size is formally defined with the variance of the individual level estimator $\hat{\theta}_k$ such that $1/n_k \propto \text{Var}(\hat{\theta}_k)$. Especially, for estimators of $\sqrt{n}$ error rate, n_k equals the number of observations for individual k. It is shown that under the settings, an individual's inference can be further improved by incorporating additional information from similar individuals, which is referred as its clique group. To be more specific, their approach aggregates individual log confidence distribution (CD) functions according to (6.4) by choosing a weight function $w_{0,k}$ which converges to an indicator function of the clique group.

6.4 Individualized Group Learning

In this section, we provide a brief introduction to the individualized group learning approach (*i*Group). Details will be discussed in Chapter 7.

Since in precision medicine (Liu and Meng, 2016; Qian and Murphy, 2011; Zhao et al., 2012) no two patients or two customers are exactly the same, heterogeneity often exists in a population. It poses a challenge to combine the data from different individuals, especially for making improved inferences in individualized learning. A class of conventional methods is to cluster/group individual entities into subgroups and, assuming homogeneity within each subgroup, then use the data in the same subgroup for statistical analysis (Jain et al., 1999; Xu and Wunsch, 2005; Agrawal et al., 1998; Binder, 1978; Ng and Han, 1994; Gan et al., 2007; Liao, 2005; Jain, 2010). The clustering and grouping in the conventional methods are typically performed in *a priori*. Such approaches have several disadvantages. Firstly, the constitution of subgroups often depends on a predetermined total number of subgroups, which is a parameter that is either difficult or not reliable to choose in practice. Secondly, since analytic outcomes and inference (e.g. estimated parameters and testing) are the same for all individuals in the same subgroup, such a procedure potentially diminishes

hidden local structures. More importantly, in many cases, there may not be clearly-cut and well-divided subgroups in the population. In these situations, the conventional subgroup analysis may impose an artificial grouping structure to the population, which can potentially lead to large biases and invalid inference for many individuals. Another class of conventional methods is to assume mixture models, including classical hierarchical models and Bayesian nonparametric models (Duda and Hart, 1973; Lindsay, 1995; Figueiredo and Jain, 2000; Ferguson, 1973; Antoniak, 1974; Lo, 1984; Teh et al., 2005). Similar to the clustering method, the mixture models assume that the population contains several homogeneous subpopulations, but unlike clustering, there is no clear boundary between the subpopulations. However, inference on each individual is not the focus of such a procedure. It is often done as an afterthought, by estimating the mixture likelihood. Furthermore, a mixture model may not be able to explain the population heterogeneity when the assumed latent structure is invalid. In addition, when given an observation, it is usually difficult to tell which subpopulation it belongs to.

In the subsequent chapters, we propose a new method called individualized group learning, abbreviated as *iGroup*. Instead of grouping at the population level, the iGroup approach focuses on each individual and forms an individualized group for the target individual, by locating individuals that share similar characteristics of the target. It sidesteps aforementioned difficulties by forming an iGroup specifically for the target individual while ignoring other entities that have little in common with the target.

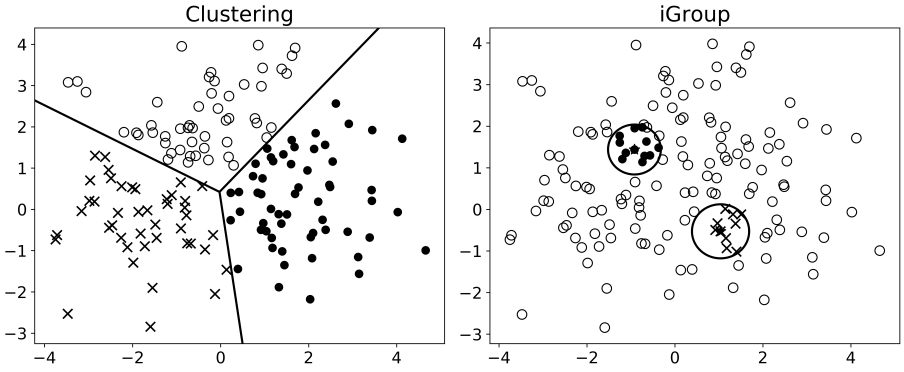


Figure 6.1: (Left) Convention clustering method divides the population into several predetermined number of groups. (Right) iGroup method find the individualized group for any given target individual.

Figure 6.1 demonstrates the difference between group identifications in a two-dimensional feature space. The left panel shows the result from a k-means clustering method with three groups. Each point is assigned with one cluster label. Data points having the same label are assumed to follow an identical statistical model, even though a large amount of heterogeneity may still exist among the individuals in the same group. The right panel demonstrates the individualized groups for two selected points (bold). Instead of assuming disjoint cluster regions, the individualized group, whose boundary is shown as a solid line, is specific and unique for each individual. Therefore, the laws for two individuals are generally different as their identified individualized groups are different. iGroup corresponds essentially to a local nonparametric approach.

There are also other methods that borrow strength from others to strengthen inference results for the target individual. A related classical approach is the k-nearest neighbor methods (k-NN) (Altman, 1992; Hall et al., 2008). The main different between the k-NN and the iGroup methods is the covariates used for identifying similarity and near neighbors. The k-NN method identifies neighborhoods usually based on covariates often without measurement errors, for example, the regressors in a regression problem. In iGroup, the covariates used for grouping, such as the exogenous variable $\boldsymbol{z}$ and the indigenous estimator $\hat{\theta}$, are both assumed to have measurement errors. Especially, the individual level point estimator $\hat{\theta}$ has never been used to measure similarity in nearest neighbor algorithms. Additionally, while the k-NN method treats every instance in the neighborhoods as equally important, the iGroup method allows different weight assignments for different individuals, which brings more flexibility. We recommend to use new weight functions to incorporate the similarity between neighbor individuals and the target one. Theoretically, when the number of individuals K approaches infinity, the radius of neighborhood identified by the k-NN method shrinks to zero as a result of bias-variance tradeoff. However, in iGroup approach, the radius of the target neighborhood does not necessarily shrink to zero, because the measurement error in $\hat{\theta}$ always exists due to finite sample size $n_k = O(1)$.

In summary, *i*Group belongs to the class of fusion learning based individualized inference approach in Section 6.3. However, in opposite to *i*Fusion approach of Shen et al. (2019), *i*Group considers a different asymptotic scheme where each individual has a finite number

of observations while the number of individuals K approaches infinity.

CHAPTER 7

Individualized Group Learning

7.1 Framework

7.1.1 Model Setup

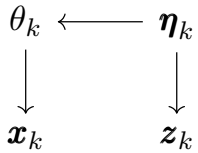
Assume for each individual $k \in \{0,1,2,\ldots,K\}$, we observe $(\boldsymbol{x}_k, \boldsymbol{z}_k)$, where observations $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ differ in their utilities. Specifically, $\boldsymbol{x}_k$ is the observed data that are directly related to the parameter of interest θ_k at the individual level, with a known distribution $\boldsymbol{x}_k \sim p(\cdot \mid \theta_k)$. The exogenous variable$\boldsymbol{z}_k$ serves as a proxy that reveals the similarity among θ's in the population level. Specifically, we assume that $\boldsymbol{z}_k$ is related to an unknown parameter $\boldsymbol{\eta}_k$ through an unknown distribution $q(\cdot;\boldsymbol{\eta}_k)$, and the parameter θ is an unknown continuous function of $\boldsymbol{\eta}$, i.e. $\theta = g(\boldsymbol{\eta})$, where the function $g(\cdot)$ is not necessarily a one-to-one mapping. The continuity of $g(\cdot)$ guarantees that closeness in $\boldsymbol{\eta}$ implies closeness in θ. The hierarchical structure and the relationship among the variables are demonstrated in Figure 7.1,

$$\theta_k \sim \pi(\cdot),$$
$$\boldsymbol{x}_k|\theta_k \sim p(\cdot;\theta_k),$$

$\boldsymbol{x}$ model

$$\theta_k = g(\boldsymbol{\eta}_k),$$
$$\boldsymbol{z}_k|\boldsymbol{\eta}_k \sim q(\cdot;\boldsymbol{\eta}_k).$$

$\boldsymbol{z}$ model



diagram

Figure 7.1: Hierarchical structure and parameter diagram.

where $\pi(\cdot)$ is an unknown (prior) population distribution of θ, which may be heterogeneous in nature. Although $\pi(\cdot)$ is unknown and unspecified, it appears in theoretical calculations throughout the theoretical analysis in this paper. Without further clarification, all

unconditioned expectations $\mathbb{E}[\cdot]$ are assumed to take over all random variables including θ_k, which follows the unknown prior $\pi(\cdot)$. Posterior expectations on θ conditioned on certain observed information are explicitly noted with π in the subscript such as $\mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0]$. The distribution $p(\cdot;\theta_k)$ is known except the parameter θ_k, but both the function $g(\cdot)$ and the distribution $q(\cdot;\cdot)$ are unknown. The role of the exogenous variable $\boldsymbol{z}_k$ will be discussed further in later sections. In some cases $\boldsymbol{z}_k$ may not be available.

One example of the above setup is that $\boldsymbol{x}_k$ is the daily stock price returns of company k, which follows a $Normal(0,\theta_k^2)$ distribution, and $\boldsymbol{z}_k$ is the company's characteristics (e.g. sectors, capital sizes, financial exposure, etc), which is related to stock volatility θ_k. Another example is that $\boldsymbol{x}_k$ is a binary indicator whether individual k has a certain disease and $\boldsymbol{z}_k$ is the individual's health indices such as weight, height, blood pressure, etc., where the underlying $\theta_k = P(\boldsymbol{x}_k = 1)$ is the probability of infection.

Denote by $C_0(\epsilon) = \{k|\tilde{d}(\theta_k,\theta_0) < \epsilon, k = 0,\ldots,K\}$ an ϵ-neighborhood (or a *clique*) of individual 0, where $\tilde{d}(\cdot,\cdot)$ is a distance/similarity measure and ϵ is the threshold value. Thus, the clique $C_0(\epsilon)$ is a set of indexes of individuals that are similar to individual 0. In our model development, we impose two regularity assumptions as below.

Assumption 7.1 (Dense Assumption)**.** *There exists a constant $d \geqslant 1$ such that for all $i = 1,\ldots,K$, $|\mathcal{C}_i(\epsilon)| \asymp K\epsilon^d$ in probability when $K \to \infty, \epsilon \to 0$.*

Assumption 7.2 (Smooth Parameter Assumption)**.** *There exists a positive constant κ, such that for all $\theta,\theta' \in \Omega_\theta$*

$$\sup_{\boldsymbol{x}} |p(\boldsymbol{x};\theta) - p(\boldsymbol{x};\theta')| \leqslant \kappa\|\theta - \theta'\|,$$

where $\|\cdot\|$ is a metric on Ω_θ.

The dense assumption suggests that individual 0 of interest is not isolated from other individuals, i.e. for arbitrarily small ϵ, there are a sufficiently large number of other individuals in its neighborhood as $K \to \infty$. The smooth parameter assumption guarantees that whenever θ and θ' are close, the distributions of $\boldsymbol{x}$ and $\boldsymbol{x}'$ induced from θ and θ', respectively, are close to each other.

Under these two assumptions, it is beneficial to aggregate information from the neighborhood to estimate θ since one can always find sufficient number of similar individuals in the neighborhood of individual θ. A key consideration in this aggregation is the familiar bias-variance trade-off — aggregation over a larger group increases the sample size thus reduces estimation variance, but it also brings bias.

7.1.2 Individualized Aggregation

Here, we provide two methods to aggregate information by creating 'pooled' estimators for θ_0. The first approach constructs a weighted estimator $\hat{\theta}_0^{(c)}(\boldsymbol{x}_0, \boldsymbol{z}_0, \mathcal{D}_x, \mathcal{D}_z)$ for the target individual 0, directly using the point estimators $\hat{\theta}_k$ of other individuals based on $\boldsymbol{x}_k$. Here we define $\mathcal{D}_x = \{\boldsymbol{x}_k\}_{k=1}^K$ and $\mathcal{D}_z = \{\boldsymbol{z}_k\}_{k=1}^K$ be all other information that is available for the individualized inference on the target individual 0.

The second approach aggregates objective functions $M_k(\theta) = M_k(\theta, \boldsymbol{x}_k)$ of other individuals, where the point estimator $\tilde{\theta}_0^{(c)}$ is obtained by optimizing an aggregated objective function. Specifically, these two methods can be formulated as

$$\text{(Aggregating estimators)} \qquad \hat{\theta}_0^{(c)} = \frac{\sum_{k=0}^K \hat{\theta}_k w(k;0)}{\sum_{k=0}^K w(k;0)}, \tag{7.1}$$

$$\text{(Aggregating objective functions)} \qquad \tilde{\theta}_0^{(c)} = \arg\min_{\theta} \sum_{k=0}^K M_k(\theta) w(k;0), \tag{7.2}$$

where $w(k;0)$ is the weight assigned to individual k when constructing iGroup estimator for individual 0.

The weight $w(k;0)$ is crucial for the aggregated estimators as it controls how much information is borrowed from other individuals. We propose to incorporate both individual level estimator $\hat{\theta}_k$ and exogenous observation $\boldsymbol{z}_k$ into the weight function as both can provide useful information of θ_0. Specifically, let

$$w(k;0) = w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0) = w_1(\boldsymbol{z}_k, \boldsymbol{z}_0) w_2(\hat{\theta}_k, \hat{\theta}_0 | \boldsymbol{z}_0, \boldsymbol{z}_k). \tag{7.3}$$

The weight is decomposed into two parts. The first part $w_1(\boldsymbol{z}_k, \boldsymbol{z}_0)$ measures the similarity

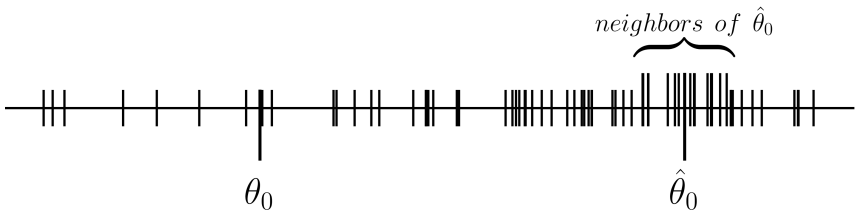


Figure 7.2: A one-dimension example in which $\hat{\theta}_0$ is away from θ_0. If one naively select individuals according to $\hat{\theta}_0$ and $\hat{\theta}_k$ directly, individuals adjacent to $\hat{\theta}_0$, but not those close to θ_0, are often selected.

between $\boldsymbol{z}_k$ and $\boldsymbol{z}_0$, and can be a kernel function

$$w_1(\boldsymbol{z}_k, \boldsymbol{z}_0) = \mathcal{K}_1\left(\frac{\|\boldsymbol{z}_k - \boldsymbol{z}\|}{b_1}\right), \tag{7.4}$$

When $\mathcal{K}_1$ has a finite support, the weight function has a hard grouping structure — individuals lying far enough from individual 0 are not considered at all. Otherwise, it has a soft grouping structure such that dissimilar individuals are assigned with non-zero but tiny weights.

The second part $w_2(\hat{\theta}_k, \hat{\theta}_0|\boldsymbol{z}_0)$ measures the similarity between $\hat{\theta}$'s. But unlike w_1, using a distance measure such as $\mathcal{K}_2(\|\hat{\theta}_k - \hat{\theta}_0\|/b_2)$ is not a good practice, since it ignores the error in $\hat{\theta}_0$ and $\hat{\theta}_k$ and $\hat{\theta}_0$ may be biased. Note that when $K \to \infty$ and $b_2 \to 0$, the kernel concentrates on a smaller and smaller area adjacent to $\hat{\theta}_0$. In this area, aggregating individual $\hat{\theta}_k$ will not improve the estimation of θ_0. An example of one-dimension case is shown in Figure 7.2. Vertical bars mark the locations of $\hat{\theta}_k$. When $\hat{\theta}_0$ is away from its target value θ_0, a small bandwidth b_2 tends to give larger weights to individuals in a local region around $\hat{\theta}_0$. Aggregating these individual $\hat{\theta}_k$ in such a local region will not correct the bias $\hat{\theta}_0 - \theta_0$.

We propose the following weight function that considers the distribution $p(\hat{\theta}|\theta)$ instead of the point estimator $\hat{\theta}$. Specifically, let

$$w_2(\hat{\theta}_k, \hat{\theta}_0|\boldsymbol{z}_0, \boldsymbol{z}_k) = \frac{\int p(\hat{\theta}_k|\theta)p(\hat{\theta}_0|\theta)p(\theta|\boldsymbol{z}_0)d\theta}{p(\hat{\theta}_k|\boldsymbol{z}_k)p(\hat{\theta}_0|\boldsymbol{z}_0)}. \tag{7.5}$$

Notice that, the posterior distribution of θ_0, given $(\hat{\theta}_0, \boldsymbol{z}_0)$, is

$$p(\theta_0|\hat{\theta}_0, \boldsymbol{z}_0) = p(\theta_0, \hat{\theta}_0|\boldsymbol{z}_0)/p(\hat{\theta}_0|\boldsymbol{z}_0) = p(\hat{\theta}_0|\theta_0)p(\theta_0|\boldsymbol{z}_0)/p(\hat{\theta}_0|\boldsymbol{z}_0).$$

If $\theta_k \equiv \theta_0$ (hence $\hat{\theta}_k$ provides useful information about θ_0), then the predictive distribution of $\hat{\theta}_k$, given $(\hat{\theta}_0, \boldsymbol{z}_0)$, is

$$p(\hat{\theta}_k|\hat{\theta}_0, \boldsymbol{z}_0) = \int p(\hat{\theta}_k|\theta)p(\theta|\hat{\theta}_0, \boldsymbol{z}_0)d\theta = \frac{\int p(\hat{\theta}_k|\theta)p(\hat{\theta}_0|\theta)p(\theta|\boldsymbol{z}_0)d\theta}{p(\hat{\theta}_0|\boldsymbol{z}_0)}.$$

Thus, the weight function $w_2(\hat{\theta}_k, \hat{\theta}_0|\boldsymbol{z}_0, \boldsymbol{z}_k)$ in (7.5) is the Radon-Nikodym derivative between the predictive distribution $p(\hat{\theta}_k|\hat{\theta}_0, \boldsymbol{z}_0)$ and the sampling distribution $p(\hat{\theta}_k|\boldsymbol{z}_k)$. As a result, for any measurable function $h(\cdot)$, we have

$$\mathbb{E}_{p(\hat{\theta}_k|\boldsymbol{z}_k)}[h(\hat{\theta}_k)w_2(\hat{\theta}_k, \hat{\theta}_0|\boldsymbol{z}_0, \boldsymbol{z}_k)] = \mathbb{E}_{p(\hat{\theta}_k|\hat{\theta}_0, \boldsymbol{z}_0)}[h(\hat{\theta}_k)].$$

That is, the weighted expectation of $h(\hat{\theta}_k)$ under the sampling distribution $p(\hat{\theta}_k|\boldsymbol{z}_k)$ equals to its expectation under the predictive distribution $p(\hat{\theta}_k|\hat{\theta}_0, \boldsymbol{z}_0)$ if $\theta_k = \theta_0$. This property brings invariance under different sampling distributions. More importantly, it shows that the weighted averages, such as (7.1) and (7.2), estimates the expectations under the predictive distribution. This gives the iGroup estimators promising asymptotic properties as we will discuss later in Section 7.2.

The shape (thin or flat) of the weight $w_2(\cdot)$ as a function of $\hat{\theta}_k$ does not change with the number of individuals K. However, the shape is influenced by the variation (accuracy) of $\hat{\theta}$. The larger the variance of $\hat{\theta}$ is, the flatter the weight function tends to be. If $\hat{\theta}_k$ is estimated without any measurement error, the weight $w_2(\hat{\theta}_k, \hat{\theta}_0|\boldsymbol{z}_0, \boldsymbol{z}_k)$ is proportional to the indicator function $I_{\{\hat{\theta}_k = \hat{\theta}_0\}}$. It reduces to the case in which the individual estimator $\hat{\theta}_0$ or the individual objective function $M_0(\theta)$ is used without grouping.

Although an unknown population distribution for θ is assumed to be $\pi(\theta)$ viewed as the prior, it does not appear explicitly in either $\hat{\theta}_0^{(c)}$ or $\tilde{\theta}_0^{(c)}$. And we'll show later in Section 3 that under mild conditions, the iGroup estimators converge to certain Bayes estimators under the unknown prior. This is similar to empirical Bayes approach (Robbins, 1956), where

the prior is unknown but a Bayes estimator is constructed. In empirical Bayes, the prior is usually estimated by either discretization or deconvolution. But our iGroup approach is different. The unknown $\pi(\theta)$ is not directly estimated. The prior information is taken into consideration by taking a (weighted) average of sample estimators or sample objective functions. And the weight w_2, which is related to $\pi(\theta)$ in a close form, is approximated by a bootstrap method in Section 7.1.3.

7.1.3 Evaluation of the Optimal Weight

The weight function $w_1(\boldsymbol{z}_k, \boldsymbol{z}_0)$ in (7.4) can be directly evaluated. Similar to a bandwidth selection problem for kernel smoothing, one can choose the bandwidth b_1 for $w_1(\boldsymbol{z}_k, \boldsymbol{z}_0)$ in (7.4) by either using the plug-in method (Chiu, 1991) or through cross-validation procedure. The plug-in bandwidth is proportional to $K^{-\frac{1}{d+4}}$ (see Section 7.2). Also, the leave-one-out cross validation process gives an empirical optimal bandwidth, as discussed in Section 7.2.6.

The evaluation of the weight function $w_2(\hat{\theta}_k, \hat{\theta}_0 \mid \boldsymbol{z}_0, \boldsymbol{z}_k)$ in (7.5) is more complicated, since the conditional probability $p(\hat{\theta}|\boldsymbol{z})$ and the integral $\int p(\hat{\theta}_0|\theta)p(\hat{\theta}_k|\theta)p(\theta|\boldsymbol{z}_0)d\theta$ are unknown as the relationship between θ and $\boldsymbol{z}$ is not explicit. We propose an approximation method to evaluate $w_2(\hat{\theta}_k, \hat{\theta}_0 \mid \boldsymbol{z}_0, \boldsymbol{z}_k)$ below.

Denote the estimator of θ_k and the observed exogenous variable $\boldsymbol{z}_k$ as the tuple $(\hat{\theta}_k, \boldsymbol{z}_k)$, $k = 0, \ldots, K$. To calculate the weight in (7.5), we treat them as $K+1$ samples from the joint distribution of $(\hat{\theta}, \boldsymbol{z})$. We use the kernel method to estimate the conditional probability $p(\hat{\theta} \mid \boldsymbol{z})$ nonparametrically by

$$\hat{p}(\hat{\theta}|\boldsymbol{z}) = \frac{\sum_{j=0}^{K} \mathcal{K}_1\left(\frac{\|\boldsymbol{z}-\boldsymbol{z}_j\|}{b_1}\right)\mathcal{K}_2\left(\frac{\|\hat{\theta}-\hat{\theta}_j\|}{b_2}\right)}{\sum_{j=0}^{K} \mathcal{K}_1\left(\frac{\|\boldsymbol{z}-\boldsymbol{z}_j\|}{b_1}\right)},$$

where $\mathcal{K}_1, \mathcal{K}_2$ are two kernel functions with b_1, b_2 as the corresponding bandwidths. To estimate the integral in (7.5), we use the interpretation discussed above that it is the conditional distribution $p(\hat{\theta}_k \mid \hat{\theta}_0, \boldsymbol{z}_0)$ given $\theta_k = \theta_0$. Hence we need samples from the joint distribution of $(\hat{\theta}, \hat{\theta}', \boldsymbol{z})$ observed from the same individual with parameter θ. However,

this is infeasible because in our problem setting, no two individual share the same true parameter θ and for each individual only one $\hat{\theta}$ is observed. To generate samples from such a distribution, we consider a bootstrap method. Denote $\hat{\theta}_k^{(1)}$ and $\hat{\theta}_k^{(2)}$ as the two bootstrap estimators for θ_k, obtained by re-sampling $\boldsymbol{x}_k$ with replacement (not applicable when $\boldsymbol{x}_k$ has few observations). Then $(\hat{\theta}_k^{(1)}, \hat{\theta}_k^{(2)}, \boldsymbol{z}_k), k = 0, \ldots, K$ is an approximate sample of $(\hat{\theta}, \hat{\theta}', \boldsymbol{z})$, guaranteeing $\hat{\theta}_k^{(1)}, \hat{\theta}_k^{(2)}, \boldsymbol{z}_k$ are generated from the same individual k. Therefore the integral can be estimated by

$$\int p(\hat{\theta}_0|\theta)p(\hat{\theta}_k|\theta)p(\theta|\boldsymbol{z}_0)d\theta \approx \frac{\sum_{j=0}^{K} \mathcal{K}_1\left(\frac{\|\boldsymbol{z}_0 - \boldsymbol{z}_j\|}{b_1}\right)\mathcal{K}_2\left(\frac{\|\hat{\theta}_0 - \hat{\theta}_j^{(1)}\|}{b_2}\right)\mathcal{K}_3\left(\frac{\|\hat{\theta}_k - \hat{\theta}_j^{(2)}\|}{b_3}\right)}{\sum_{j=0}^{K} \mathcal{K}_1\left(\frac{\|\boldsymbol{z}_0 - \boldsymbol{z}_j\|}{b_1}\right)},$$

where $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ are three kernel functions with b_1, b_2, b_3 as the corresponding bandwidths. The bandwidths can be selected by either minimizing asymptotic mean integrated squared error (AMISE) or a rule-of-thumb bandwidth estimator. This estimation of the integral is an approximation that requires K to be sufficiently large.

7.2 Theoretical Analysis

In this section, we consider several model settings for which we apply the proposed *i*Group method and discuss their corresponding theoretical properties, especially in terms of their asymptotic performance. In particular, we first define a target estimator Θ_0 that minimizes the Bayes risk, and then investigate the asymptotic performance of iGroup estimators in (7.1) and (7.2) in approximating the target estimator Θ_0. We also quantify the bias and variance of iGroup estimators as well as the target estimator Θ_0 in term of estimating θ_0. Particularly, in this chapter we consider the asymptotic framework that the number of individuals K goes to infinity, while the number of observations for each individual n is fixed and finite.

7.2.1 Risk Decomposition and the Target Estimator

We are interested in making inference about individual 0, with given data information $\mathcal{D}_x, \mathcal{D}_z$ that may include the observations $\boldsymbol{x}_0$ and $\boldsymbol{z}_0$ plus information from other relevant individuals. Let $\delta_0(\mathcal{D}_x, \mathcal{D}_z)$ be a point estimator for θ_0, which is constructed with information sets $\mathcal{D}_x$ and $\mathcal{D}_z$. The iGroup estimator $\hat{\theta}_0^{(c)}$ in (7.1) is such an estimator. Similarly, $\delta_0(\mathcal{D}_x)$ and $\delta_0(\mathcal{D}_z)$ are point estimators constructed solely based on either $\mathcal{D}_x$ or $\mathcal{D}_z$. Under squared loss, the overall risk of δ_0 in estimating θ_0 can be decomposed into two non-negative parts: the expected squared error of δ_0 in estimating the corresponding posterior mean and the overall risk of the posterior mean itself, as shown in Proposition 7.1.

Proposition 7.1. *Suppose θ_0 has a prior distribution $\pi(\cdot)$. Under squared loss, we have the following overall risk decomposition.*

$$\begin{aligned}
\mathbb{E}[(\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \theta_0)^2] &= \mathbb{E}[(\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0])^2] + \mathbb{E}[(\mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0] - \theta_0)^2], \\
\mathbb{E}[(\delta_0(\mathcal{D}_x) - \theta_0)^2] &= \mathbb{E}[(\delta_0(\mathcal{D}_x) - \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0])^2] + \mathbb{E}[(\mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0] - \theta_0)^2], \\
\mathbb{E}[(\delta_0(\mathcal{D}_z) - \theta_0)^2] &= \mathbb{E}[(\delta_0(\mathcal{D}_z) - \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0])^2] + \mathbb{E}[(\mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0] - \theta_0)^2],
\end{aligned}$$

where $\mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0]$, $\mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0]$ and $\mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0]$ are the posterior means under prior $\pi(\cdot)$ and observations $(\boldsymbol{x}_0, \boldsymbol{z}_0)$, $\boldsymbol{x}_0$ and $\boldsymbol{z}_0$ correspondingly.

The proof is given in Appendix A.

Proposition 7.1 reveals that the overall risk is minimized by setting δ_0 to the corresponding posterior mean under the prior $\pi(\cdot)$, which is the population-level (unknown) distribution for θ_0. Throughout this paper, we call the estimator that minimizes the overall risk the *target estimator*. More specifically, under squared loss and different information sets, we denote the target estimators with

$$\Theta_0(\boldsymbol{x}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0], \quad \Theta_0(\boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0] \text{ and } \Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0]. \quad (7.6)$$

Here, ℓ_2 refers to the squared loss. For the ease of presentation, we also use a simple notation Θ_0 to represent one of the Bayes estimators in (7.6) when its meaning is apparent.

Similarly, for a general loss function $L(\hat{\theta},\theta)$, we define the target estimator as the Bayes estimator that minimizes the expected loss, given the available observation on individual 0 and the prior $\pi(\cdot)$ such that

$$\begin{aligned}\Theta_0(\boldsymbol{x}_0;L) &= \arg\min_{\delta}\mathbb{E}_{\pi}[L(\delta,\theta_0)\mid \boldsymbol{x}_0],\\ \Theta_0(\boldsymbol{z}_0;L) &= \arg\min_{\delta}\mathbb{E}_{\pi}[L(\delta,\theta_0)\mid \boldsymbol{z}_0],\\ \Theta_0(\boldsymbol{x}_0,\boldsymbol{z}_0;L) &= \arg\min_{\delta}\mathbb{E}_{\pi}[L(\delta,\theta_0)\mid \boldsymbol{x}_0,\boldsymbol{z}_0].\end{aligned} \tag{7.7}$$

A similar risk decomposition is demonstrated in Proposition 7.2 below. Again, for the ease of notation, we simply use Θ_0 to represent one of the Bayes estimators in (7.7) when its meaning is apparent.

Proposition 7.2. *Suppose θ_0 has a prior distribution $\pi(\cdot)$ and $L(\hat{\theta},\theta)$ is a loss function, which is second-order partially differentiable with respect to $\hat{\theta}$ such that $L'(\hat{\theta},\theta)=\partial L/\partial\hat{\theta}$ and $L''(\hat{\theta},\theta)=\partial^2 L/\partial\hat{\theta}^2$. Then for the estimator δ_0 constructed based on information set $\mathcal{D}_x$, $\mathcal{D}_z$ or $(\mathcal{D}_x,\mathcal{D}_z)$, we have*

$$\mathbb{E}[L(\delta_0,\theta_0)]=\frac{1}{2}\mathbb{E}[L''(\Theta_0,\theta_0)(\delta_0-\Theta_0)^2]+\mathbb{E}[L(\delta_0,\theta_0)]+o(\mathbb{E}[(\delta_0-\Theta_0)^2]),$$

where Θ_0 is the corresponding Bayes estimator based on the same information set as δ_0.

The proof is given in Appendix A.

The target estimator Θ_0 as a function of $\boldsymbol{x}_0$ and $\boldsymbol{z}_0$ is not directly available, because neither the population distribution $\pi(\theta_0)$ nor the likelihood function $p(\boldsymbol{z}_0\mid\theta_0)$ is explicitly known or assumed. The iGroup estimator $\hat{\theta}_0^{(c)}$ in (7.1) constructed based on observed finite sample $\mathcal{D}_x,\mathcal{D}_z$ is desired to approach the target estimator Θ_0 when more and more similar individuals contribute to the estimator $\hat{\theta}_0^{(c)}$. See Diaconis and Freedman (1986) for discussions of target point estimators and target parameters in Bayesian literature.

7.2.2 Case 1: With Exogenous Variable z Only

In the cases when the individual level estimator $\hat{\theta}_k$ is not reliable to construct the individual groups, iGroup may be constructed with the exogenous variable $\boldsymbol{z}$ only. In this case, the corresponding target estimator is defined as:

$$\Theta_0(\boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0], \tag{7.8}$$

where $p(\theta_0 \mid \boldsymbol{z}_0) \propto p(\boldsymbol{z}_0 \mid \theta_0)\pi(\theta_0)$. Although $\boldsymbol{x}_0$ is not used for grouping and thus does not appear in (7.8), the data $\mathcal{D}_x$ is used in iGroup estimators in (7.1) and (7.2). Recall that the relationship between θ_k and $\boldsymbol{\eta}_k$ is given by a deterministic relationship

$$\theta_k = g(\boldsymbol{\eta}_k), \quad \text{for } k = 0, 1, \ldots, K, \tag{7.9}$$

where $g(\cdot)$ is an unknown continuous function. Furthermore, $\boldsymbol{z}_k$ is a noisy observation of $\boldsymbol{\eta}_k$. Since $\boldsymbol{\eta}$ is a conceptual parameter, we may simply assume that

$$\boldsymbol{z}_k = \boldsymbol{\eta}_k + \epsilon_k, \quad \text{for } k = 0, \ldots, K,$$

where the error satisfies $\mathbb{E}(\epsilon_k) = 0$, $\mathrm{Var}(\epsilon_k) = \sigma_z^2 \boldsymbol{\Sigma}_z$ with $\|\boldsymbol{\Sigma}_z\| = 1$.

Suppose $\hat{\theta}_k$ is an unbiased estimator of θ_k. Then, the combined estimator

$$\hat{\theta}_0^{(c)} = \frac{\sum_{k=0}^K \mathcal{K}\left(\dfrac{\|\boldsymbol{z}_k - \boldsymbol{z}_0\|}{b}\right)\hat{\theta}_k}{\sum_{k=0}^K \mathcal{K}\left(\dfrac{\|\boldsymbol{z}_k - \boldsymbol{z}_0\|}{b}\right)} \tag{7.10}$$

has all the properties of a conventional kernel smoothing estimator if $\mathcal{K}$ is a standard kernel function. The boundary and asymptotic conditions/assumptions on the weight function $\mathcal{K}$ and the bandwidth b are summarized in Assumption 7.3.

Assumption 7.3 (Boundary and asymptotic conditions)**.** *The kernel function $\mathcal{K}(\cdot)$ satisfies*

$$\mathcal{K} \geqslant 0, \quad \int |\mathcal{K}(u)| du < \infty, \quad \lim_{|u| \to \infty} u\mathcal{K}(u) \to 0.$$

And, in addition, when $K \to \infty$, b *satisfies* $b \to 0, \quad b^d K \to \infty$.

Theorem 7.1. *Under the conditions in Assumption 7.1 - 7.3, we have*

$$\hat{\theta}_0^{(c)} \longrightarrow \Theta_0(\boldsymbol{z}_0; \ell_2) \quad \textit{in probability.}$$

The optimal choice of the bandwidth is $\hat{b} \asymp K^{-1/(d+4)}$ *such that the optimal MSE is* $\mathbb{E}[(\hat{\theta}_0^{(c)} - \Theta_0)^2] \asymp K^{-4/(d+4)}$.

Theorem 7.1 follows immediately from consistency theorem on a standard multivariate kernel smoothing estimator (Wasserman, 2010). When the number of individuals K goes to infinity, the bias of $\hat{\theta}_0^{(c)}$ with bandwidth b is of order b^2 and the variance is of order $(b^d K)^{-1}$, where d is the dimension of $\boldsymbol{z}$ as defined in Assumption 7.1. In such case, the asymptotic optimal choice of bandwidth that minimizes the mean squared error, $b^4 + (b^d K)^{-1}$, is of order $K^{-1/(d+4)}$, same as a d-dimensional kernel smoothing problem.

Another way of combining individuals is aggregating the objective functions as shown in (7.2). A combined estimator with respect to kernel $\mathcal{K}(\cdot)$ is defined by

$$\tilde{\theta}_0^{(c)} = \arg\min_{\theta} \sum_{k=0}^{K} \mathcal{K}\left(\frac{\|\boldsymbol{z}_k - \boldsymbol{z}_0\|}{b}\right) M_k(\theta).$$

The estimator is consistent and has a similar asymptotic performance to a d-dimensional kernel smoothing estimator as stated in Theorem 7.2. This approach is useful especially when $\hat{\theta}_k$ is not available, such as in the cases that the number of observations for each individual is less than the number of parameters.

Theorem 7.2. *Suppose the conditions in Assumption 7.3 hold and in addition,*

1. $M_k(\theta)$ *is convex and second order partial differentiable with respect to* θ,
2. *for any given* θ, $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}}[\dfrac{\partial M_{\boldsymbol{x}}(\theta)}{\partial \theta}]$ *as a function of* $\boldsymbol{z}$ *is continuous,*
3. $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}_0}[M_{\boldsymbol{x}}(\theta)]$ *has a unique minimum at* $\theta = \Theta_0(\boldsymbol{z}_0; \ell_2)$.

Then

$$\tilde{\theta}_0^{(c)} \longrightarrow \Theta_0 \quad \textit{in probability.}$$

The optimal choice of bandwidth b is $\hat{b} \asymp K^{-1/(d+4)}$ and the optimized mean squared error is $\mathbb{E}[(\hat{\theta}_0^{(c)} - \Theta_0)^2] \asymp K^{-4/(d+4)}$.

The proof is given in Appendix A.

The above theorems suggest that the individualized combined estimator by aggregating either individual estimators $\hat{\theta}_k$ or objective functions $M_k(\theta)$ would result in an improvement in mean squared error and it shares a similar asymptotic performance as a d-dimensional kernel smoothing estimator.

When $\sigma_z = 0$, $\Theta_0(\boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0] \equiv \theta_0$. Hence, estimating Θ_0 becomes estimating the unknown function $g(\cdot)$ evaluated at $\boldsymbol{z}_0$. When $\sigma_z > 0$, Θ_0 and θ_0 are in general different. Let B_0 and V_0 be the bias and variance of the target estimator $\Theta_0(\boldsymbol{z}_0; \ell_2)$ in estimating θ_0 such that

$$B_0(\theta_0) := \mathbb{E}_{\theta_0}[\Theta_0(\boldsymbol{z}_0; \ell_2)] - \theta_0, \quad V_0(\theta_0) = Var_{\theta_0}[\Theta_0(\boldsymbol{z}_0; \ell_2)]. \tag{7.11}$$

The above bias and variance are defined with respect to a fixed θ_0 with random $\boldsymbol{z}_0$.

Theorem 7.3. *The asymptotic bias and variance of $\hat{\theta}_0^{(c)}$ in estimating a fixed θ_0 are given by*

$$\begin{aligned} \mathbb{E}_{\theta_0}[\hat{\theta}_0^{(c)}] - \theta_0 &= B_0(\theta_0) + O_p(b^2), \\ Var_{\theta_0}[\hat{\theta}_0^{(c)}] &= V_0(\theta_0) + O_p\left(\frac{1}{Kb^d}\right), \end{aligned}$$

where the intrinsic bias B_0 and the intrinsic variance V_0 are defined in (7.11).

The proof is given in Appendix. In the conditional probabilities, $\Theta_0 = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0]$, as a function of $\boldsymbol{z}_0$, is considered random under a given θ_0.

The bias and variance of $\hat{\theta}_0^{(c)}$ in terms of estimating a fixed θ_0 can therefore be decomposed into two parts. The first part (the intrinsic part) comes from the bias and variance of estimating $\Theta_0[\boldsymbol{z}_0]$ itself to θ_0 and the second part comes from estimating Θ_0 nonparametrically. Since $\boldsymbol{z}$ is observed with error, this is similar to error in variable problem where certain intrinsic bias cannot be avoided (Fuller, 2009; Carroll et al., 1995; Wansbeek and Meijer, 2000; Bound et al., 2001). Such intrinsic bias and variance are asymptotically linear

of σ_z^2, which is the noise level of $\boldsymbol{z}_k$, as shown in Theorem 7.4. Especially, when σ_z^2 is exactly zero, all intrinsic terms vanish, and it reduces to the exact case when $\Theta_0 = \theta_0$.

Theorem 7.4. *Suppose $g(\cdot)$ is second-order differentiable and the distribution of ϵ_k has finite higher moments. Then, for a fixed θ_0, when $\sigma_z^2 \to 0$,*

$$B_0 \asymp \sigma_z^2, \quad V_0 \asymp \sigma_z^2.$$

The proof is given in Appendix A.

Research in nonparametric regression with error in variable shows a slower convergence rate to recover the function $\theta_0 = g(\boldsymbol{\eta})$ at any given $\boldsymbol{\eta}$ (Stefanski and Carroll, 1990; Fan and Truong, 1993). Our problem is different. We focus on providing a point estimator of $\theta_0 = g(\boldsymbol{\eta}_0)$ without knowning $\boldsymbol{\eta}_0$, but its noisy version $\boldsymbol{z}_0$. Even if we known the function $g(\cdot)$ precisely, θ_0 is not known as we do not observe η_0.

When considering an individual with fixed but unobserved $(\theta_0, \boldsymbol{\eta}_0)$, it is difficult to choose an optimal bandwidth by bias-variance optimization with the non-zero intrinsic terms in Theorem 7.3, because in this case the asymptotic mean squared error $(B_0 + O_p(b^2))^2 + V_0 + O_p((Kb^d)^{-1})$ may not have a local minimum. However, if we assume the target individual 0 is randomly chosen from the population, the target estimator Θ_0 is the estimator that minimizes the overall risk under squared loss, i.e. a Bayes estimator, because it minimizes the squared loss pointwise for any $\boldsymbol{z}_0$. Furthermore, immediately from Theorem 7.1, $\hat{\theta}_0^{(c)}$ is a consistent estimator for Θ_0. The overall performance of $\hat{\theta}_0^{(c)}$ for all individuals of the population could be optimized by choosing a proper bandwidth b as stated in the following Theorem 7.5. It provides a way to optimize the bandwidth globally.

Theorem 7.5. *Assume Assumption 7.1 - 7.3 hold, then the estimator $\hat{\theta}_0^{(c)}$ has the following Bayes risk under squared loss*

$$\mathbb{E}[(\hat{\theta}_0^{(c)} - \theta_0)^2] = R_0 + O_p(b^4) + O_p\left(\frac{1}{Kb^d}\right),$$

where

$$R_0 = Var[\Theta_0 - \theta_0]$$

is the risk of the Bayes estimator $\Theta_0 = E_\pi[\theta|\boldsymbol{z}_0]$*, and all above expectations is taken over all random variables assuming an empirical population distribution* $\pi(\cdot)$ *for* θ_0*. The optimal choice of the bandwidth* b *is* $b \asymp K^{1/(d+4)}$ *with the corresponding overall risk* $R_0 + O_p(K^{4/(d+4)})$*.*

The proof is given in Appendix A.

The magnitude of the measurement error of $\boldsymbol{z}_k$, measured by σ_z^2, compared to that of the individual estimation error is crucial for the performance of the iGroup method. The bias and variance of iGroup estimator increase when σ_z^2 increases (see Theorem 7.4). And the asymptotic Bayes risk R_0 also depends on σ_z^2. When iGroup is based on unreliable $\boldsymbol{z}$, it could result in a worse estimator compared to the one without any grouping. This phenomenon will be demonstrated in Section 7.3.

Results in Theorems 7.3, 7.4 and 7.5 can be generalized to the iGroup estimator $\tilde{\theta}_0^{(c)}$, which combines the objective functions, except that the target estimator changes from $\mathbb{E}_\pi[\theta|z_0]$ is replaced by $\arg\min_\theta \mathbb{E}_\pi[M(\theta)|z_0]$. As shown in (A.1) in the Appendix, $\tilde{\theta}_0^{(c)}$ is asymptotically a kernel smoothing estimator with the same bias and variance rates.

7.2.3 Case 2: Without Exogenous Variables

In this case, we assume the exogenous variable $\boldsymbol{z}$ is not available. Our target estimator is $\Theta_0(\boldsymbol{x}; \ell_2) = \mathbb{E}_\pi[\theta_0|\boldsymbol{x}_0]$ under squared loss and is $\Theta_0(\boldsymbol{x}_0; L) = \arg\min_\theta \mathbb{E}_\pi[L(\theta, \theta_0) \mid \boldsymbol{x}_0]$ under a general loss function L. The iGroup estimation depends solely on $\hat{\theta}$. The weight function (7.5) used in (7.1) and (7.2) now reduces to

$$w_2(\hat{\theta}_k, \hat{\theta}_0) = \frac{\int p(\hat{\theta}_k|\theta)p(\hat{\theta}_0|\theta)\pi(\theta)d\theta}{\int p(\hat{\theta}_k|\theta)\pi(\theta)d\theta \int p(\hat{\theta}_0|\theta)\pi(\theta)d\theta}, \tag{7.12}$$

where $\pi(\theta)$ corresponds to the unknown distribution of θ in the whole population. As discussed in Section 7.1.3, an estimation of this weight function can be achieved by kernel density estimation on the bootstrapped samples $(\hat{\theta}_k^{(1)}, \hat{\theta}_k^{(2)})$.

The weight function (7.12) is used to aggregated individual unbiased estimators to the posterior mean, and to aggregate objective functions $M : \Omega_\theta \times \Omega_\theta \to \mathbb{R}$ to the corresponding Bayes estimator under certain loss function, as shown in Theorems 7.6 and 7.7.

Theorem 7.6. *Suppose $w_2(\hat{\theta}_k, \hat{\theta}_0)$ is defined as in Equation (7.12) and $\hat{\theta}_k$ is a sufficient and unbiased estimator of θ_k for all k, then as $K \to \infty$:*

$$\hat{\theta}_0^{(c)} \to \Theta_0(\boldsymbol{x}_0; \ell_2) \quad \textit{in probability.}$$

Furthermore, if $\mathbb{E}_{\hat{\theta}_0}[w_2^2(\hat{\theta}_k, \hat{\theta}_0)] < \infty$ for any fixed $\hat{\theta}_0$ and $\mathbb{E}_\pi[\hat{\theta}^2] < \infty$, then

$$\sqrt{K}(\hat{\theta}_0^{(c)} - \Theta_0) = O_p(1).$$

The proof is given in Appendix A.

For the aggregated estimator (7.2), suppose the objective function $M : \Omega_\theta \times \Omega_\theta \to \mathbb{R}$ used satisfies

$$\int M(\theta, \hat{\theta}) p(\hat{\theta}|\theta') d\hat{\theta} = L(\theta, \theta') + C(\theta'), \tag{7.13}$$

where L is non-negative and $L(\theta, \theta) = 0$ for all θ, and C is constant with respect to θ. Then L is the loss function corresponding to M, under which the target estimator is

$$\Theta_0(\boldsymbol{x}_0; L) = \arg\min_\theta \int L(\theta, \theta_0) p(\hat{\theta}_0|\theta_0) \pi(\theta_0) d\theta_0.$$

For example, if the objective function M is the negative log-likelihood function $M(\theta, \hat{\theta}) = -\log p(\hat{\theta}|\theta)$, then the corresponding loss function $L(\theta, \theta')$ is the Kullback-Leibler divergence of the given parameters.

Theorem 7.7. *If for any given $\hat{\theta}$, $M(\theta, \hat{\theta})$ as a function of θ is convex and second-order differentiable, then the combined estimator $\tilde{\theta}_0^{(c)}$ using the objective function M converges in probability to the target estimator under the loss function L as $K \to \infty$:*

$$\hat{\theta}_0^{(c)} = \arg\min_\theta \sum_{k=0}^{K} w_2(\hat{\theta}_k, \hat{\theta}_0) M(\theta, \hat{\theta}_k) \xrightarrow{P} \Theta_0(\boldsymbol{x}_0; L).$$

Furthermore, if $\mathbb{E}_{\hat{\theta}_0}[w_2(\hat{\theta}_k, \hat{\theta}_0) M'_\theta(\theta_0, \hat{\theta})]^2 < \infty$ for any fixed $\hat{\theta}_0$,

$$\sqrt{K}(\tilde{\theta}_0^{(c)} - \Theta_0) = O_p(1).$$

The proof is given in Appendix A.

The finite second moment conditions in Theorems 7.6 and 7.7 are satisfied in most cases. Both Theorems 7.6 and 7.7 assume an accurate estimation of the weight $w_2(\hat{\theta}_k, \hat{\theta}_0)$ (with an error rate smaller than $O_p(K^{-1/2})$. With the accurate weights $w_2(\hat{\theta}_k, \hat{\theta}_0)$, both iGroup estimators have faster convergence rates to the target estimator Θ_0 than the nonparametric one in Theorems 7.1.

When no accurate estimations for $w_2(\hat{\theta}_k, \hat{\theta}_0)$ are feasible, we proposed an approximate estimator for $w_2(\hat{\theta}_k, \hat{\theta}_0)$ in Section 7.1.3, using a set of bootstrap samples $(\hat{\theta}_k^{(1)}, \hat{\theta}_k^{(2)})$ for $k = 0, \ldots, K$. When $\boldsymbol{z}$ is not available, the integral $\int p(\hat{\theta}_k|\theta)p(\hat{\theta}_0|\theta)\pi(\theta)d\theta$ can be estimated by a kernel density estimator in a lower dimensional space:

$$\frac{1}{K+1}\sum_{j=0}^{K}\mathcal{K}_1\left(\frac{|\hat{\theta}_j^{(1)} - \hat{\theta}_k|}{b_1}\right)\mathcal{K}_2\left(\frac{|\hat{\theta}_j^{(2)} - \hat{\theta}_0|}{b_2}\right),$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are two kernel functions with b_1, b_2 the corresponding bandwidths. The bootstrap estimation of the weight $w_2(\hat{\theta}_k, \hat{\theta}_0)$ has a nonparametric error rate $O_p(K^{-1/(d'+2)})$, where d' is the dimension of θ_0. This inaccuracy gives rise to the final error rate in Theorem 7.6 and 7.7 such that for $\hat{\theta}_0^{(c)}$ (or $\tilde{\theta}_0^{(c)}$) constructed based on $\hat{w}_2(\hat{\theta}_k, \hat{\theta}_0)$ with error rate $O_p(K^{-1/(d'+2)})$, $\hat{\theta}_0^{(c)} - \Theta_0(\boldsymbol{x}_0; \ell_2) = O_p(K^{-1/(d'+2)})$ and $\tilde{\theta}_0^{(c)} - \Theta_0(\boldsymbol{x}_0; L) = O_p(K^{-1/(d'+2)})$. Both are slower than $O_p(K^{-1/2})$.

The performance of the target estimator $\Theta_0(\boldsymbol{x}_0; \ell_2)$ in estimating θ_0 strongly depends on the accuracy of individual level $\hat{\theta}_k$. Define the bias and variance of the target estimator $\Theta_0(\boldsymbol{x}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0]$ by

$$B_0(\theta_0) = \mathbb{E}_{\theta_0}[\Theta_0(\boldsymbol{x}_0; \ell_2)] - \theta_0, \quad V_0(\theta_0) = \mathrm{Var}_{\theta_0}[\Theta_0(\boldsymbol{x}_0; \ell_2)]. \tag{7.14}$$

Suppose $\hat{\theta}_0 = \theta_0 + \zeta_0$ with $\mathbb{E}[\zeta_0] = 0$ and $\mathbb{E}[\zeta_0^2] = \sigma_\theta^2$. Similar to Theorem 7.4, B_0 and V_0 are of order σ_θ^2 when $\sigma_\theta^2 \to 0$.

Theorem 7.8. *Suppose ζ_0 has finite higher moments. Then, when $\sigma_\theta^2 \to 0$, the bias and*

variance of the target estimator $\Theta_0(\boldsymbol{x}_0;\ell_2)$ *with respect to a fixed* θ_0 *are*

$$B_0 \asymp \sigma_\theta^2, \quad V_0 \asymp \sigma_\theta^2,$$

where B_0 *and* V_0 *are defined in* (7.14).

The proof is provided in Appendix A.

When $\hat{\theta}_0$ is exact such that $\sigma_\theta = 0$, the target estimator equals to the true parameter θ_0 as the weight function $w_2(\hat{\theta}_k,\hat{\theta}_0)$ assigns zero weight for all other individuals except individual 0. Similar results hold for the target estimator $\Theta_0(\boldsymbol{x}_0;L)$.

7.2.4 Case 3: The Complete Case

When both $\hat{\theta}$ and $\boldsymbol{z}$ are available and reasonably accurate, we should use both information to improve the inference via grouping. Assuming $\hat{\theta}$ is sufficient for θ_0, the target estimator is $\Theta_0(\boldsymbol{x}_0,\boldsymbol{z}_0;\ell_2) = \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0,\boldsymbol{z}_0]$ under squared loss and $\Theta_0(\boldsymbol{x}_0,\boldsymbol{z}_0;L) = \arg\min_\theta \mathbb{E}_\pi[L(\theta,\theta_0) \mid \hat{\theta}_0,\boldsymbol{z}_0]$ under other loss function L. The following results are based on a combination of both information.

Theorem 7.9. *Suppose* $\hat{\theta}_k$ *is a sufficient and unbiased estimator for* θ_k*, and* $\hat{\theta}_0^{(c)}$ *is a combined estimator as in (7.1) with the weight functions (7.3), (7.4) and (7.5), where* $\mathcal{K}(\cdot)$ *is a kernel function satisfying Assumption 7.3. Then under Assumptions (7.1) and (7.2)*

$$\hat{\theta}_0^{(c)} \to \Theta_0(\boldsymbol{x}_0,\boldsymbol{z}_0;\ell_2) \quad \textit{in probability.}$$

With the optimal bandwidth $\hat{b}$ *chosen to be* $\hat{b} \asymp K^{1/(d+4)}$*, the optimal mean squared error is* $\mathbb{E}[\hat{\theta}_0^{(c)} - \Theta_0]^2 \asymp K^{-4/(d+4)}$.

The proof is given in Appendix A.

Let $M(\theta,\hat{\theta})$ be the corresponding objective function as defined in (7.13). We have that the aggregated estimator (7.2) based on the objective function $M(\theta,\hat{\theta})$ converges to the target estimator $\Theta_0(\boldsymbol{x}_0,\boldsymbol{z}_0;L)$ as shown in the following Theorem 7.10.

Theorem 7.10. *If for any given $\hat{\theta}$, $M(\theta, \hat{\theta})$ as a function of θ is convex and second-order differentiable, then under Assumptions (7.1) and (7.2), the combined estimator $\tilde{\theta}^{(c)}$ using the objective function M satisfying (7.13) converges to the target estimator:*

$$\tilde{\theta}_0^{(c)} = \arg\min_{\theta} \sum_{k=1}^{K} w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0) M(\theta, \hat{\theta}_k) \xrightarrow{P} \Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; L).$$

With the optimal bandwidth $\hat{b}$ chosen to be $\hat{b} \asymp K^{1/(d+4)}$, the optimal mean squared error is $\mathbb{E}[\tilde{\theta}_0^{(c)} - \Theta_0]^2 \asymp K^{-4/(d+4)}$.

The proof is given in Appendix A.

Define the bias and variance of the target estimator $\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2)$ as

$$B_0(\theta_0) = \mathbb{E}_{\theta_0}[\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2)] - \theta_0, \quad V_0(\theta_0) = \mathrm{Var}_{\theta_0}[\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2)]. \tag{7.15}$$

The asymptotic rate of B_0 and V_0 as σ_θ^2 or σ_z^2 approaches zero is shown in Theorem 7.11.

Theorem 7.11. *Suppose $g(\cdot)$ is second order differentiable and ϵ_k and ζ_k have finite higher moments. If B_0 and V_0 are as defined in (7.15), then*
(i) for a fixed σ_z^2, when $\sigma_\theta^2 \to 0$,

$$B_0 \asymp \sigma_\theta^2, \quad V_0 \asymp \sigma_\theta^2.$$

(ii) for a fixed σ_θ^2, when $\sigma_z^2 \to 0$,

$$B_0 \asymp \sigma_z^2, \quad V_0 \asymp \sigma_z^2.$$

The proof is provided in Appendix A.

The bias and variance of the target estimator is of the order of the more accurate one between $\boldsymbol{z}_0$ and $\hat{\theta}_0$. Especially, when either is exact such that $\sigma_z^2 = 0$ or $\sigma_\theta^2 = 0$, the target estimator equals the true parameter θ_0.

7.2.5 Further Results on Risk Decomposition

Let $\hat{\theta}_0^{(c)}$ be an iGroup estimator as defined in (7.1) based on information sets $\{\boldsymbol{z}\}$, $\{\hat{\theta}\}$ or $\{\hat{\theta}, \boldsymbol{z}\}$ as in Sections 7.2.2, 7.2.3 and 7.2.4, respectively. Let Θ_0 be the target estimator in any of the three cases: $\Theta_0(\boldsymbol{x}_0; \ell_2)$, $\Theta_0(\boldsymbol{z}_0; \ell_2)$ or $\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2)$, depending on the information set used in $\hat{\theta}_0^{(c)}$. We have $\hat{\theta}_0^{(c)} \to \Theta_0$ in probability. When both $\hat{\theta}$ and $\boldsymbol{z}$ are available for all individuals, the overall risk of $\hat{\theta}_0^{(c)}$ under the prior $\pi(\theta)$ can be decomposed into three components as shown in Proposition 7.3 as an extension to Proposition 7.1.

Proposition 7.3. *Suppose $\hat{\theta}_0^{(c)}$ is an iGroup estimator as defined in (7.1) with the target estimator Θ_0. Then*

$$R(\hat{\theta}_0^{(c)}) = R_{np}(\hat{\theta}_0^{(c)}) + R_{target}(\Theta_0),$$

where $R(\hat{\theta}_0^{(c)}) = \mathbb{E}[(\hat{\theta}_0^{(c)} - \theta_0)^2]$ is the overall risk of $\hat{\theta}_0^{(c)}$ under squared loss and prior $\pi(\theta_0)$, and

$$R_{np}(\hat{\theta}_0^{(c)}) = \mathbb{E}[(\hat{\theta}_0^{(c)} - \Theta_0)^2], \quad R_{target}(\Theta_0) = \mathbb{E}[(\Theta_0 - \theta_0)^2]$$

are the risk components from the nonparametric estimation and the target estimator itself, respectively.

Furthermore, assuming both $\boldsymbol{x}$ and $\boldsymbol{z}$ are available, for $\Theta_0 = \Theta_0(\boldsymbol{x}_0; \ell_2)$ or $\Theta_0 = \Theta_0(\boldsymbol{z}_0; \ell_2)$, which only uses partial information, we have

$$R_{target}(\Theta_0) = R_{inf}(\Theta_0) + R_0,$$

where $R_{inf}(\Theta_0) = \mathbb{E}[(\Theta_0 - \Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2))^2]$ is the risk premium resulting from using partial information, and $R_0 = \mathbb{E}[(\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2) - \theta_0)^2]$ is the overall risk of $\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2)$.

The proof is provided in Appendix A.

The decomposition in Proposition 7.3 reveals a guideline to optimize the iGroup estimator. The overall risk of iGroup estimator $\hat{\theta}_0^{(c)}$ can be decomposed into two parts: one from the nonparametric estimation of the target estimator and the other from the risk of the target estimator itself. The risk component R_{np} involves the bandwidth b in the nonparametric estimation. The corresponding optimal bandwidth is chosen as in a high-dimensional kernel

smoothing problem (see Theorems 7.1, 7.5 and 7.9), since the bandwidth does not appear in the other risk terms.

The risk component R_{target} evaluates the performance of the target estimator. Different choices in constructing iGroup weight correspond to different Θ_0's. Such difference is revealed by decomposing R_{target} into two parts: R_{inf} is the risk term arising from using partial information and R_0 is the risk of the target estimator $\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2)$, which incorporates the full information set. Since R_{inf} obtains its minimum at $\Theta_0 = \Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2)$, it is always (asymptotically) optimal to use the full information set $\{\hat{\theta}, \boldsymbol{z}\}$ in grouping, if both are available as in the complete case. On the other hand, if $\hat{\theta}$ (or $\boldsymbol{z}$) is extremely noisy such that $\Theta_0 = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0] \approx \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]$ (or $\Theta_0 = \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0] \approx \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]$, respectively), it is more practical to use $\boldsymbol{z}$ only (or $\hat{\theta}$ only, respectively) for grouping, since it will have similar performance but less computational cost, and finite sample variation.

The last risk component R_0 is the minimum overall risk one can achieve. In our approach, such a minimum risk can be asymptotically reached when both $\hat{\theta}$ and $\boldsymbol{z}$ are included in grouping and the number of individuals K approaches infinity. When $\hat{\theta}$ or $\boldsymbol{z}$ is exact, $\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 | \hat{\theta}_0, \boldsymbol{z}_0] = \theta_0$ and R_0 is 0. In this case, all iGroup estimators in (7.1) converges to θ_0. The three risk components of different iGroup models are compared in Table 7.1. Note that the rate of R_{np} for Case 2 assumes an accurate evaluation of the weight function $w_2(\hat{\theta}_k, \theta_0)$.

	iGroup Set	R_{np}	R_{target}	
			R_{inf}	R_0
Case 1	$\{\boldsymbol{z}\}$	$\asymp K^{-4/(d+4)}$	> 0	same value
Case 2	$\{\hat{\theta}\}$	$\asymp K^{-1}$	> 0	
Case 3	$\{\hat{\theta}, \boldsymbol{z}\}$	$\asymp K^{-4/(d+4)}$	$= 0$	

Table 7.1: Comparison of the three risk components in different iGroup cases.

Similar to Proposition 7.3, the risk decomposition for the iGroup estimator $\tilde{\theta}_0^{(c)}$ in (7.2) is provided in Proposition 7.4 as an extension to Proposition 7.2.

Proposition 7.4. *Suppose the loss function L is as defined in (7.13). The iGroup estimator $\tilde{\theta}_0^{(c)}$ is defined in (7.2) with the target estimator Θ_0. If $L(\hat{\theta}, \theta)$ is second-order partially*

differentiable with respect to $\hat{\theta}$ *such that* $L'(\hat{\theta},\theta) = \partial L/\partial\hat{\theta}$ *and* $L''(\hat{\theta},\theta) = \partial^2 L/\partial\hat{\theta}^2$*, then*

$$\tilde{R}(\tilde{\theta}_0^{(c)}) = \tilde{R}_{np}(\tilde{\theta}_0^{(c)}) + \tilde{R}_{target}(\Theta_0) + o(\mathbb{E}[(\tilde{\theta}_0^{(c)} - \Theta_0)^2]),$$

where $\tilde{R}(\tilde{\theta}_0^{(c)}) = \mathbb{E}[L(\tilde{\theta}_0^{(c)},\theta_0)]$ *is the overall risk of* $\tilde{\theta}_0^{(c)}$ *under loss* L *and prior* $\pi(\theta)$*, and*

$$\tilde{R}_{np}(\tilde{\theta}_0^{(c)}) = \frac{1}{2}\mathbb{E}[L''(\Theta_0,\theta_0)(\tilde{\theta}_0^{(c)} - \Theta_0)^2],\ \tilde{R}_{target}(\Theta_0) = \mathbb{E}[L(\Theta_0,\theta_0)],$$

are the risk components from the nonparametric estimation of the target estimator and the target estimator itself, respectively.
Furthermore, assuming both $\boldsymbol{x}$ *and* $\boldsymbol{z}$ *are available, for any* $\Theta_0 = \Theta_0(\boldsymbol{z}_0; L)$ *or* $\Theta_0 = \Theta_0(\boldsymbol{x}_0; L)$*, which only uses partial information, we have*

$$\tilde{R}_{target}(\Theta_0) = \tilde{R}_{inf}(\Theta_0) + \tilde{R}_0,$$

where $\tilde{R}_0 = \mathbb{E}[L(\Theta_0(\boldsymbol{x}_0,\boldsymbol{z}_0;L),\theta_0)]$ *denotes the overall risk of* $\Theta_0(\boldsymbol{x}_0,\boldsymbol{z}_0;L)$ *and* $\tilde{R}_{inf}(\Theta_0) = \mathbb{E}[L(\Theta_0,\theta_0)] - \tilde{R}_0$ *is the risk premium resulting from using partial information.*

The proof is given in Appendix A.

7.2.6 Bandwidth Selection

For real applications, the bandwidth b in the weight function (7.4) remains to be tuned. Ideally one would perform bandwidth selection to the target individual θ_0. However, cross validation cannot be implemented to determine b with only one estimator $\hat{\theta}_0^{(c)}$ for a single individual. Instead, we consider a set Ω_0 around target individual 0 such that the bandwidth b is tuned to minimize the averaged risk over Ω_0.

When Ω_0 is chosen as the full set $\{1,2,\ldots,K\}$, it is the global bandwidth selection scheme that usually used in kernel smoothing and machine learning. However, the bandwidth selected by such global optimization is not optimal for the particular target individual 0. A cross validation set Ω_0 localized to individual 0 is more appreciated to tune this individualized local bandwidth. When tuning the bandwidth in w_1 over $\boldsymbol{z}_k$'s, such a set Ω_0

can be constructed based on $\boldsymbol{z}_0$ such as $\Omega_0(\boldsymbol{z}_0, \epsilon) = \{k \in \{1, \ldots, K\} : \|\boldsymbol{z}_0 - \boldsymbol{z}_k\| \leqslant \epsilon\}$.

Suppose $\hat{\theta}_k$'s are available and the individual estimators are aggregated to form an iGroup estimator as described in (7.1). The goal is to choose a bandwidth b that minimizes the local risk function over Ω_0 (under squared loss) around θ_0

$$R_{\Omega_0}(b) = \mathbb{E}\left[\frac{1}{|\Omega_0|}\sum_{k\in\Omega_0}(\hat{\theta}_k^{(c)} - \theta_k)^2\right].$$

The cross-validation error we use is computed as

$$CV_{\Omega_0}(b) = \frac{1}{|\Omega_0|}\sum_{k\in\Omega_0}\left(\hat{\theta}_{(-k)}^{(c)} - \hat{\theta}_k\right)^2,$$

where $\hat{\theta}_{(-k)}^{(c)}$ is the leave-one-out estimator defined by

$$\hat{\theta}_{(-k)}^{(c)} = \frac{\sum_{l\neq k}\hat{\theta}_l w(l;k)}{\sum_{l\neq k} w(l;k)}. \tag{7.16}$$

It is worth to point out that although the cross validation set Ω_0 is localized/individualized, the leave-one-out estimators (7.16) still utilize all individuals instead of limited to Ω_0.

It is seen in Proposition 7.5 that the leave-one-out cross-validation can estimate the local risk over Ω_0 up to a constant and hence be useful.

Proposition 7.5. *Suppose $\hat{\theta}_k$ is an unbiased estimator for θ_k for all $k = 1, \ldots, K$ and the weight function $w(l;k)$ satisfies*

$$\frac{w(k;k)}{\sum_{l\neq k} w(l;k)} = O\left(\frac{1}{K}\right). \tag{7.17}$$

Then

$$\mathbb{E}[CV_{\Omega_0}(b)] = R_{\Omega_0}(b) + C_{\Omega_0} + O\left(\frac{1}{K}\right),$$

where C_{Ω_0} is related to Ω_0 but is a constant with respect to b.

The proof is given in Appendix A.

A sufficient condition for the weight function to satisfy (7.17) is that the function is

bounded. With bounded weights, we have

$$\frac{w(k;k)}{\sum_{l\neq k} w(l;k)} \to \frac{w(k;k)}{K\mathbb{E}w(\cdot;k)} = O\left(\frac{1}{K}\right).$$

Common kernels such as the boxed, Gaussian and Epanechnikov kernels satisfy this condition. Our choice of weight function (7.5) with a bounded kernel $\mathcal{K}$ satisfies the condition as well.

Similar results hold for aggregating objective functions (7.2) as long as the objective function is convex and second-order differentiable, and a Taylor series expansion is available.

Beside the theoretical discussions on *i*Group's asymptotic performance, there are many other factors that may affect the accuracy in real applications with finite number of individuals. First of all, the weight component $w_2(\cdot)$ is estimated from bootstrapped samples. It lowers the convergence rate since bootstrapped samples from finite population are usually correlated. Secondly, computing the full weight function requires a kernel density estimation in a high dimensional space. When K is finite, aggregating individuals with weights evaluated directly from a high dimensional space suffers from the lack of sample size. It often requires some feature selection procedures to reduce the dimension.

Therefore, when the weight estimation is not accurate and when the sample size is limited, the complete case may not be the best choice. In real application, we suggest using (local) cross-validation to tune the bandwidth and to choose the most appropriate weight formulation.

7.3 Simulations and Examples

7.3.1 Simulation: Noisy Exogenous Variables

In this example, the performance of using an exogenous variable z in iGroup is studied (corresponding to Case 1 in Section 7.2.2). Suppose, for each individual, the true parameter θ is a quadratic function of η:

$$\theta_k = g(\eta_k) = (\eta_k + 1)^2.$$

The relationship is set to a quadratic form because a continuous function of z can be approximated by a quadratic function within a small enough neighborhood of z_0. A population of size $K = 1000$ is generated with their η_k's following a Gaussian distribution $N(0.2, 1)$. For each individual k, let $\hat{\theta}_k$ be a sufficient unbiased estimator of θ_k using $\boldsymbol{x}_k$ such that $\hat{\theta}_k$ is directly generated with error $\epsilon \sim N(0, \tau^2 = 1)$ and there is no need to generate $\boldsymbol{x}_k$ explicitly. z_k is a noisy observation of η_k such that $z_k \sim N(\eta_k, \sigma^2)$.

More specifically, the dataset is generated by the following hierarchical structure.

$$\eta_k \sim N(0.2, 1), \quad \theta_k = (\eta_k + 1)^2, \quad \hat{\theta}_k \sim N(\theta_k, 1), \quad z_k \sim N(\eta_k, \sigma^2),$$

for $k = 1, \ldots, K$.

The estimator in (7.10) is used by setting $\mathcal{K}(\cdot)$ to the Gaussian kernel.

The parameter σ^2 controls the noise level in the observed z_k. Both individualized performances at $\theta_0 = 1$ and the overall performance over the population are studied at the six choices of noise levels $\sigma = 0, 0.2, 0.4, 0.6, 0.8, 1.0$ with 1000 replications each.

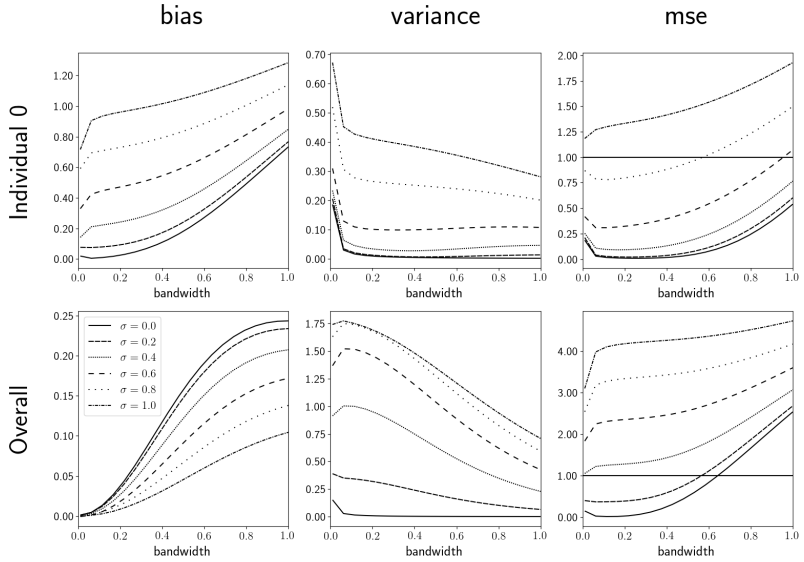


Figure 7.3: Bias, variance and mean squared error as a function of bandwidth under different noise levels for individual 0 (top) and the population (bottom)

The in-sample performance of the iGroup estimators are demonstrated in Figure 7.3. The first row shows the bias, variance and mean squared error for the individual at $\theta_0 = 1$,

while the second row plots the overall performance by averaging the individual performance over the population. Every curve represents a performance measure (bias, variance or MSE) as a function of the bandwidth b used in weight calculation in (7.4) and six different curves distinguish different noise levels σ^2.

From Figure 7.3, it is seen that an increase in the noise level in $\boldsymbol{z}_k$ increases both the bias and variance of the iGroup estimator. When $\sigma > 0$, an intrinsic bias is observed for individual 0 when the bandwidth shrinks to zero, while at the population level, the average bias vanishes when the bandwidth shrinks to zero as the iGroup estimator converges to the target estimator $\Theta_0(\boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0]$, whose expectation is $\mathbb{E}_\pi[\theta_0]$. Recall that the individual estimate $\hat{\theta}_k$ without grouping has a risk $\tau^2 = 1.0$ by the simulation design. It is marked on the right panels by the horizontal line. When the noise level σ exceeds 0.4, both the individual level and population level risk are worse than using $\hat{\theta}_k$ directly without grouping. Smaller noise in z_k would significantly reduce the risk of the iGroup estimator.

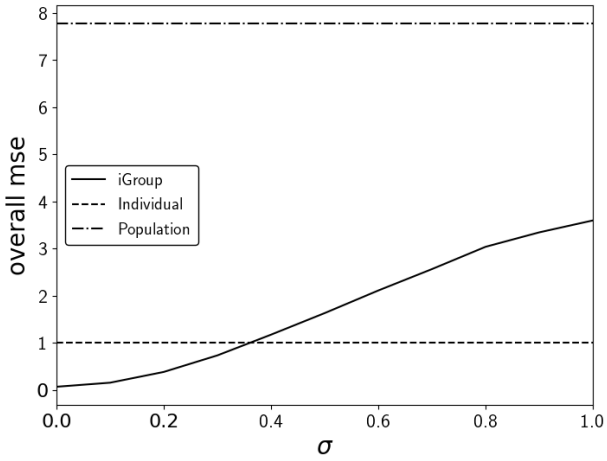


Figure 7.4: Overall MSE of three estimators: individual level, iGroup with cross validation and population level.

In real applications, the performance plots such as Figure 7.3 are not available without knowing the true parameter. As suggested in Section 7.2.6, an optimal bandwidth can be selected by leave-one-out cross validation. We simply use the global set $\Omega_0 = \{1, \ldots, K\}$ to tune the bandwidth. Figure 7.4 compares the mean square errors of three different estimators under different noise level settings for σ^2. The individual level estimator uses $\hat{\theta}_k$,

which achieves a constant MSE at $\tau^2 = 1$. The population level estimator uses the averaged estimator $(\sum_{k=1}^{K} \hat{\theta}_k)/K$, assuming population homogeneity. The iGroup estimator uses the estimator (7.10) and selects the optimal bandwidth by leave-one-out cross validation over a grid of bandwidths. The population level estimator is always the worst because the homogeneity population assumption is invalid in this simulation. The overall MSE of the iGroup estimator is a monotone increasing function of the noise level σ, because the intrinsic bias and variance increase with σ. The iGroup estimator outperforms the individual estimator when σ is below the threshold $\sigma = 0.35$. It also suggests that the iGroup method works better when more accurate exogenous variable z is used.

7.3.2 Simulation: Short Time Series

In this simulation study, the individualized grouping learning method is applied to a set of short time series without any exogenous information, corresponding to Case 2 in Section 7.2.3. Suppose we have $K = 200$ time series following an AR(1) model. Their AR coefficients $\theta_1, \ldots, \theta_{200}$ are drawn randomly from a beta-shaped distribution on $[-1, 1]$ such that

$$\frac{\theta_k + 1}{2} \sim Beta(4, 4), \quad k = 1, \ldots, 200. \tag{7.18}$$

The length of each time series is 10. They are generated from their stationary distributions:

$$x_{k,0} \sim N\left(0, \frac{\sigma^2}{1 - \theta_k^2}\right),$$
$$x_{k,t} = \theta_k x_{k,t-1} + \epsilon_{k,t}, \quad k = 1, \ldots, 200, \ t = 1, \ldots, 10,$$

where $\epsilon_{k,t} \sim N(0, \sigma^2)$ and $\sigma = 3$.

Four estimators are used and their mean squared errors averaged over the 200 individual time series are compared. The individual level estimator is based on each time series of 10 observations and does not borrow any information from the others. It is an unbiased estimator for each individual. The iGroup1 estimator aggregates the log-likelihood functions according to (7.2), where the weight function used is (7.12), which is estimated by bootstrap samples. The bootstrap estimates are obtained based on multinomial samples of

(x_{t-1}, x_t) pairs for each individual. The bandwidth used in estimating $w_2(\hat{\theta}_k, \hat{\theta}_0)$ in (7.12) is chosen by cross-validation as in a kernel density estimation problem. The iGroup2 estimator aggregates individual level estimators by the weight function in Equation (7.12), the same weight function as in the iGroup1 estimator. These three methods do not utilize the true prior distribution. The fourth estimator, the oracle one, uses the posterior mean as the estimator with the true population prior (7.18) as the prior. The oracle estimator, which is the best point estimator for θ_0 given the prior information $\pi(\cdot)$, is the target estimator $\Theta_0(\boldsymbol{x}_0; \ell_2)$ for iGroup methods.

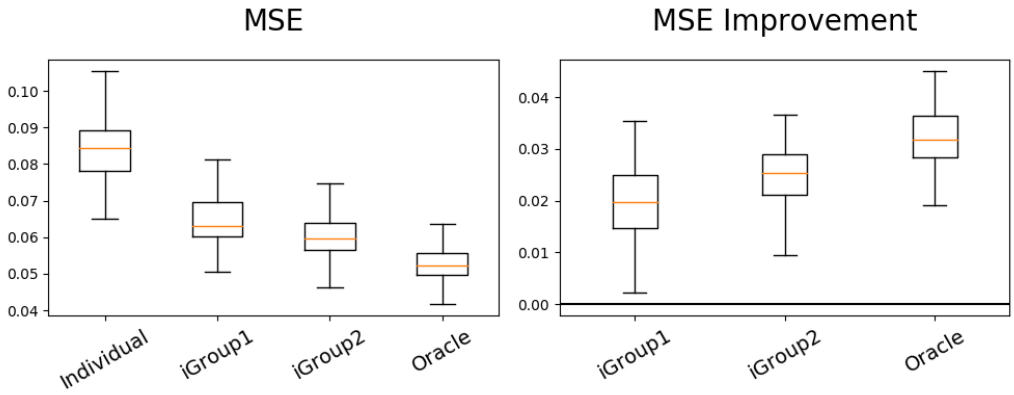


Figure 7.5: Comparison of the averaged MSE over 200 individuals on 100 replications for four estimators

The simulation (including generating the data) is repeated 100 times. The box plots of the mean squared errors of the four estimators are reported in the left panel of Figure 7.5. On average, the iGroup1 and iGroup2 estimators achieve smaller mean squared errors and smaller variances compared with the individual one. The oracle estimator is the best among those four with the smallest average error and variation. The iGroup estimators are quite close to the oracle one. The slight worse performance is due to the approximation error when constructing the weight functions. Between the two iGroup estimators, iGroup2 is slightly better than iGroup1 because the loss function used in iGroup2 is the squared loss, whose overall risk is minimized by aggregating $\hat{\theta}_k$ (See Theorem 7.6).

The right panel in Figure 7.5 plots the improvement (difference) of the mean square errors of the iGroup estimators and the oracle estimator over the individual estimator for the 100 replications. It shows that in all experiment replications, the mean square errors

of the iGroup estimators are uniformly better than the individual one. Estimation does benefit from individualized grouping in this case.

7.3.3 Simulation: Complete Case

In this simulation, we compare the performance of different iGroup estimators constructed on different information sets when both $\hat{\theta}$ and $\boldsymbol{z}$ are available as in Case 3 discussed in Section 7.2.4. Consider a population with $n = 1024$ individuals following:

$$\eta_k \sim N(0,1), \quad \theta_k = \sin(\pi\eta_k), \quad z_k \sim N(\eta_k, \sigma^2), \quad x_{k,1}, x_{k,2}, \ldots, x_{k,n} \sim N(\theta_k, \sigma_x^2),$$

for $k = 1, \ldots, 1024$. θ is the parameter of interest. Individual estimator used is

$$\hat{\theta}_k = \frac{1}{n}\sum_{i=1}^{n} x_{k,i} \text{ for } k = 1, \ldots, 1024.$$

Four approaches are investigated here as special cases of the iGroup method. iGroup($\emptyset$) is the individual estimation without grouping, i.e. using $\hat{\theta}_k$ as the estimator. iGroup(z) uses the exogenous observation z only for grouping and an iGroup estimator is obtained by aggregating $\hat{\theta}$'s using $w_1(\boldsymbol{z}_k, \boldsymbol{z}_0)$ in (7.4), where the bandwidth b is selected by leave-one-out cross validation. iGroup($\hat{\theta}$) uses $\hat{\theta}_k$ only for grouping, using $w_2(\hat{\theta}, \hat{\theta}')$ in (7.5) as the weight function. The weight is approximated by kernel density estimation on the bootstrapped samples with bandwidth selected by cross validation. And lastly, iGroup(z, $\hat{\theta}$) uses both z and $\hat{\theta}$ for calculating the weight function $w(\boldsymbol{z}_k, \hat{\theta}_k; \boldsymbol{z}_0, \hat{\theta}_0)$ in (7.3) as discussed in Section 7.2.4, with the bandwidth selected by leave-one-out cross validation.

Several different (n, σ, σ_x) configurations are studied. The mean square errors are reported in Table 7.2. The smallest MSE across the different methods is shown in bold face for each configuration. From Table 7.2, it is seen that in Configurations 6 to 11, using both $\boldsymbol{z}$ and $\hat{\theta}$ outperforms the other three methods. However, it is worth to point out that it is not always the best. When z is relatively accurate and $\hat{\theta}$ is not so as in Configurations 1, 2, 3 and 5, using $\boldsymbol{z}$ alone is better than involving $\hat{\theta}$ in the grouping. The reason is that the weight function used in the estimation is an approximation based on bootstrap sampling, which is

$Config.$	n	$\tau^2 = \sigma_x^2/n$	σ	iGroup($\emptyset$)	iGroup($\hat{\theta}$)	iGroup(z)	iGroup(z, $\hat{\theta}$)
1	5	0.20	0.10	0.200	0.163	**0.044**	0.154
2	5	0.20	0.15	0.200	0.163	**0.090**	0.163
3	5	0.20	0.20	0.200	0.163	**0.137**	0.170
4	5	0.20	0.30	0.200	**0.163**	0.200	0.179
5	10	0.10	0.10	0.100	0.089	**0.048**	0.059
6	10	0.10	0.15	0.100	0.089	0.089	**0.070**
7	10	0.10	0.20	0.100	0.089	0.099	**0.077**
8	10	0.10	0.30	0.100	0.089	0.100	**0.084**
9	20	0.05	0.10	0.050	0.046	0.044	**0.040**
10	20	0.05	0.15	0.050	0.046	0.050	**0.044**
11	20	0.05	0.20	0.050	0.046	0.050	**0.045**
12	20	0.05	0.30	0.050	**0.046**	0.050	0.047

Table 7.2: Mean squared error for different configurations.

not accurate when the sample size n is too small (as discussed in Section 7.2.6). It is also intuitive since using inaccurate $\hat{\theta}_k$ for grouping may reduce the grouping quality. When $\boldsymbol{z}$ is quite noisy as in Scenario 4 and 12, using $\hat{\theta}$ only is better than using the complete information set. Note that when the bandwidth in $w_1(\boldsymbol{z}_k, \boldsymbol{z}_0)$ shrinks to zero, iGroup(z) reduces to the individual estimator and the complete estimator iGroup(z, $\hat{\theta}$) reduces to iGroup($\hat{\theta}$). However, due to the randomness from finite sample size and possible overfitting, iGroup($\hat{\theta}$) or iGroup(z) sometimes performs better.

In conclusion, we suggest the following brief guideline in choosing iGroup models. When $\hat{\theta}$ is relatively inaccurate and the bootstrap method has unignorable error, it is better not to use $\hat{\theta}$ in grouping. When $\boldsymbol{z}$ is relatively inaccurate, it is better to either use $\hat{\theta}$ only or use the full model. But when using the full model, the bandwidth needs to be tuned carefully around zero. When both $\hat{\theta}$ and $\boldsymbol{z}$ are considerably accurate, it is beneficial to consider both in grouping.

7.3.4 Example: Value at Risk of Stock

In this example we use iGroup to improve the estimation of Value at Risk in stock returns. Denote the return of stock k in day t as $r_{t,k}$. The one-day value at risk (VaR) of $r_{t,k}$, denoted as $\widehat{VaR}_{t,k}$, is defined as the smallest quantity v such that the probability of the event $r_{t,k} \leqslant -v$ is no greater than a predetermined confidence level α (for example, 1%).

Statistically, $-v$ is the α quantile of $r_{t,k}$. VaR is widely used in quantitative finance and risk management to estimate the possible losses in worse cases (e.g. 1% lower quantile) due to adverse market moves. In practice, it is usually difficult to estimate the value of risk because it requires a large size of data to estimate small quantiles accurately, but the market conditions change over time, which limits the available sample size. In this application, we consider the daily return of 490 stocks in S&P 500 for 2016. Three approaches to estimate VaR are compared.

Individual VaR estimation using empirical quantiles: A naive method to estimate VaR is to use the empirical quantile of $r_{t-1,k}, \ldots, r_{t-S,k}$. When α is set to be 1% and $S = 100$, we have $\widehat{VaR}(t,k) = \min\{r_{t-1,k}, r_{t-2,k}, ..., r_{t-100,k}\}$. Such a quantile estimation is not very accurate. On one hand, when S is small and there is not enough observations, the empirical quantile is not defined. On the other hand, S cannot be very large as the market changes over time and so does the distribution of returns.

Market Level VaR: The second approach assumes homogeneity among all stocks. The value-at-risk could then be estimated by pooling historical returns of all stocks. In this case, the estimator is

$$\widehat{VaR}(t,k) = Q_\alpha\left(\bigcup_{l=1}^{K}\bigcup_{s=1}^{S}\{r_{t-s,l}\}\right),$$

where $Q_\alpha(A)$ is the empirical α quantile estimator given a set of observations A. Pooling observations from other stocks bring a significant bias if the homogeneity assumption is not valid.

iGroup Estimation: The third approach is an application of the iGroup learning method. Assume on each day, each stock return follows the Fama-French three factor model (Fama and French, 1993):

$$\begin{aligned} r_{t,k} &= \alpha_{t,k} + r_f + b_{0,t,k}(MKT_t - r_f) + b_{1,t,k}SMB_t + b_{2,t,k}HML_t + \epsilon_{t,k}, \\ \epsilon_{t,k} &\sim \mathcal{N}(0, \sigma_k^2), \end{aligned}$$

where MKT, SMB and HML are the three Fama-French factors, and $b_{0,k,t}$, $b_{1,k,t}$ and $b_{2,k,t}$ are the corresponding coefficients for the stock labeled k at time t. The three coefficients

characterize stocks by their sensitivity to the corresponding factors. In this model, we assume the Fama-French coefficients b_0, b_1, b_2 vary over time slowly. Therefore, the Fama-French coefficients could be used as the exogenous variable $\boldsymbol{z}$ in our iGroup framework. To be more specific, the iGroup estimator is

$$\widehat{VaR}(t,k) = Q_\alpha^{(w)}\left(\bigcup_{l=1}^{K}\bigcup_{s=1}^{S}\{(r_{t-s,l}, w(\boldsymbol{z}_{t,l};\boldsymbol{z}_{t,k}))\}\right),$$

where $Q_\alpha^{(w)}(\cdot)$ is the empirical α quantile estimator from a weighted sample and $\boldsymbol{z}_{t,k} = (b_{0,t,k}, b_{1,t,k}, b_{2,t,k})$ are the Fama-French coefficients of stock k fitted using the returns in the S days before day t. The weight function here is chosen to be a Gaussian kernel

$$w(\boldsymbol{z}_{t,l};\boldsymbol{z}_{t,k}) \propto \exp\left(-\frac{\|\boldsymbol{z}_{t,l}-\boldsymbol{z}_{t,k}\|_2^2}{2b^2}\right).$$

The bandwidth b is the parameter to be tuned. Although the iGroup approach pools all other stocks just as the market level method, it assigns different weights to different stocks based on the similarity of characteristics of the stocks, e.g. the Fama-French coefficients in our case. The market level estimator can be viewed as an extreme case of iGroup estimation when the bandwidth b approaches ∞. The individual estimator is another extreme when the bandwidth b shrinks to 0. Note that, the weighted empirical quantile function used in iGroup estimation is equivalent to aggregating the following objective function

$$M_k(\theta;t) = \sum_{s=1}^{S}|r_{t-s,k}-\theta|\left(\alpha\mathbf{1}_{\{r_{t-s,k}>\theta\}} + (1-\alpha)\mathbf{1}_{\{r_{t-s,k}\leqslant\theta\}}\right)$$

by the weight $w_1(\boldsymbol{z}_k, \boldsymbol{z}_0)$ in (7.4).

In this study, we use $\alpha = 0.01$, $S = 100$, and $K = 490$. The prediction error is measured over 250 trading days in the year 2016 for 490 stocks using

$$RMSE = \left[\frac{1}{490}\sum_{k=1}^{490}\left(\frac{1}{250}\sum_{t=1}^{250}\mathbf{1}_{\left\{r_{t,k}\leqslant\widehat{VaR}(t,k)\right\}} - 0.01\right)^2\right]^{1/2},$$

where $\widehat{VaR}(t,k)$ is based on returns $\{r_{t-1,k},\ldots,r_{t-100,k}, k = 1,\ldots,490\}$.

Figure 7.6 shows the RMSE curve as a function of the bandwidth b. The bandwidth

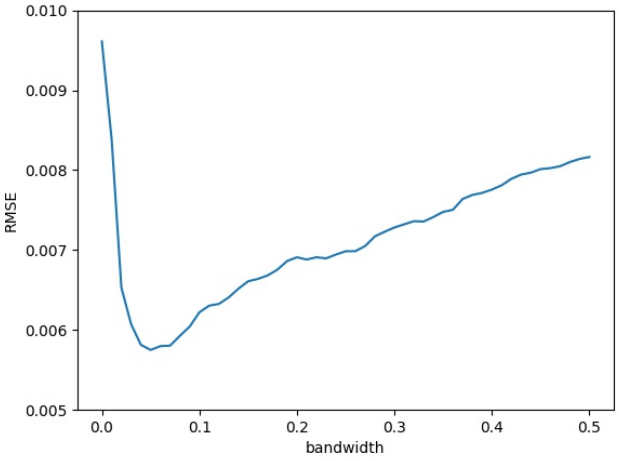


Figure 7.6: Prediction error (RMSE) as a function of bandwidth.

controls the bias-variance tradeoff. It is seen from the figure that the V-shaped RMSE curve decreases at the beginning and achieves a minimal value at approximately $b = 0.05$ with minimum RMSE being 5.75×10^{-3}. The RMSEs of each model are shown in Table 7.3. The iGroup estimator improves the accuracy significantly.

Method	Individual Estimation	Market Estimation	iGroup Estimation
RMSE	9.61×10^{-3}	1.34×10^{-2}	5.75×10^{-3}

Table 7.3: Prediction error for three candidate models.

7.3.5 Example: Maritime Anomaly Detection

The maritime transportation system is critical to the U.S. and world economy. For security and environmental concerns, it is important to have an efficient detection and risk assessment system for maritime traffic over space and time. Automatic Identification System (AIS) is an automatic tracking system and are mandatory installed on ships such that the maritime information, including GPS location, speed, heading, etc., is reported periodically. The global AIS system receives data from approximately a million ships with updates for each ship as frequently as every two seconds while in motion and every three minutes while at anchor. The data are available at https://marinecadastre.gov/ais/.

In this example, we focused on 534 voyages of tankers and cargo vessels arriving at

the Port of Newark between July and November 2014. We investigated their approaching behaviors starting from crossing the 12 nautical mile US territorial sea (TS) boundary to arriving at the port. Two features are considered in this study: the trajectory and the sailing time (duration). The trajectory, treated as an exogenous variable $\boldsymbol{z}$, is a polygonal line consisting of a sequence of reported GPS locations during the approach. The 534 approaching trajectories are plotted in Figure 7.7 along with the coastlines around the Port of Newark. The sailing time, treated as the observation x_k, is the time spent in the approaching procedure starting at the time of entering the 12 nautical miles territorial sea of U.S. and ending at one of the docks in the Port of Newark. Our goal is to identify outliers in sailing time given the trajectory. In this case the parameter of interest is the mean and standard deviation of sailing time, $\theta_k = (\mu_k, \sigma_k)$, such that an outlier can be identified by two standard deviation rule, i.e. individual k is an outlier in time if $|x_k - \hat{\mu}_k| \geqslant 2\sigma_k$.

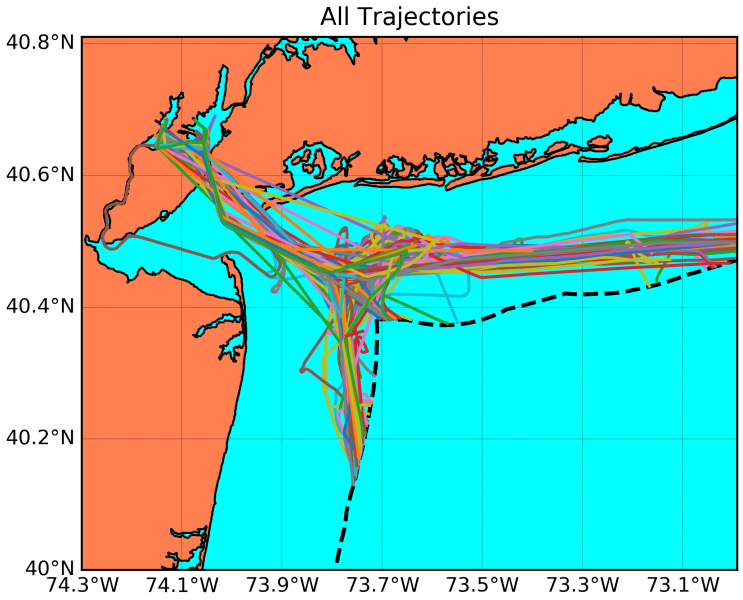


Figure 7.7: All 534 trajectories approaching the Port of Newark

The trajectory is a functional feature that requires special treatment. Every trajectory consists of a sequence location reports ordered in time. Since the reporting intervals are irregular, it cannot be considered as a 2-dimensional regular time series of equal time intervals. However, since we utilize the trajectory as an exogenous variable $\boldsymbol{z}$ in the iGroup

framework, we only need a proper distance/similarity measure defined for any trajectory pairs. Here, we use the dynamic time warping (DTW) distance as the similarity measure. Dynamic time warping is widely used as a similarity measure between two time series for studies in speech recognition and other applications (Sakoe and Chiba, 1978; Juang, 1984; Nakagawa and Nakanishi, 1988; Koenig et al., 2008). It finds the optimal monotone one-to-one mapping between two sequences such that the average pairwise distance is minimized.

For simplicity, for each individual voyage, we use its nearest 40 neighbors in terms of DTW to form iGroups with equal weight. Figure 7.8 shows four typical trajectories (top) and their individualized groups identified by its DTW neighbors (bottom).

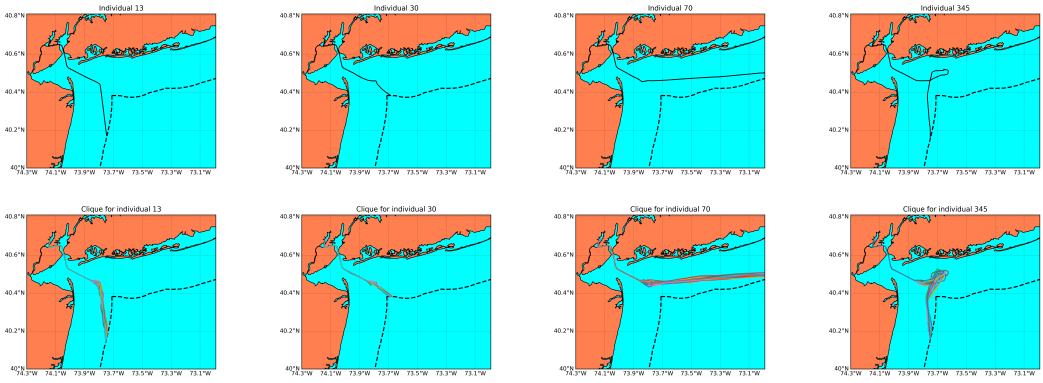


Figure 7.8: Four typical trajectories and their identified individualized groups.

Since the individual level estimator for θ_k is not available as we only have one observation x_k per individual. The iGroup estimator is constructed by aggregating the log-likelihood functions. In this case, it is equivalent to estimate θ_k by the sample mean and the sample standard deviation from the formed igroup. Since our main focus is to identify outliers, we exclude the target from the estimation. Denote $\mathcal{C}_k$ as the individualized group (clique) identified by the DTW distance for voyage k. Note that we control $|\mathcal{C}_k| = 40$. The iGroup estimator can be constructed as

$$\mu_k^{(c)} = \frac{\sum_{i \in \mathcal{C}_k} x_i}{|\mathcal{C}_k|}, \quad \sigma_k^{(c)} = \frac{\sum_{i \in \mathcal{C}_k} (x_i - \mu_k^{(c)})^2}{|\mathcal{C}_k| - 1}.$$

Then the risk score (the likelihood of being an outlier) of individual k can be obtained as

$$1-2P\left(Z>\left|\frac{x_k-\mu_k^{(c)}}{\sigma_k^{(c)}}\right|\right),$$

where $Z\sim N(0,1)$.

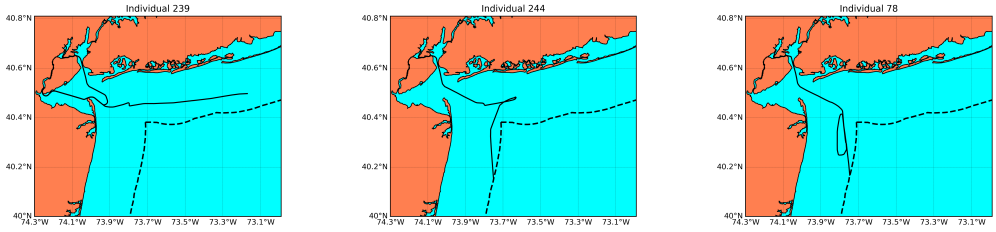


Figure 7.9: Outliers among vessels/voyages in trajectories of vessels heading to Port of Newark

In these 534 vessels, 95 outliers with risk scores larger than 95% were determined as abnormal. A manual inspection reveals that they belong to three categories (with some overlaps between (a) and (b)): (a) 40 vessels had a prior dock before the Port of Newark (left panel of Figure 7.9); (b) 18 vessels were anchored somewhere outside the port for an extremely long time (middle panel); (c) the other 43 vessels were traveling too fast/slow compared with their iGroup (right panel). Figure 7.9 shows typical trajectories of the three categories. Due to the limited population, vessels with few similar trajectories are also classified as abnormal such as the one shown in the right panel in Figure 7.9.

PART III

Kronecker Product Approximation

CHAPTER 8

Kronecker Product Decomposition

8.1 Kronecker Product

The Kronecker product, denoted by "$\otimes$", is a binary operator so that the Kronecker product of two matrices results in a larger block matrix containing all cross products of the elements in the two component matrices. The Kronecker product has wide applications in signal processing, image restoration, quantum computing and many other scientific researches. For example, in the statistical model for a multi-input multi-output (MIMO) channel communication system, Werner et al. (2008) modeled the covariance matrix of channel signals as the Kronecker product of the transmit covariance matrix and the receive covariance matrix. In compressed sensing, Duarte and Baraniuk (2012) utilized Kronecker products to provide a sparse basis for high-dimensional signals. In image restoration, Kamm and Nagy (1998) considered the blurring operator as a Kronecker product of two smaller matrices. In quantum computing, Kaye et al. (2007) represented the joint state of quantum bits as a Kronecker product of their individual states.

We first give the definition of the Kronecker product.

Definition 8.1. *The Kronecker product of a $p \times q$ real matrix $\boldsymbol{A}$ and a $p' \times q'$ real matrix $\boldsymbol{B}$ is a $(pp') \times (qq')$ real matrix given by*

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & a_{1,2}\boldsymbol{B} & \cdots & a_{1,q}\boldsymbol{B} \\ a_{2,1}\boldsymbol{B} & a_{2,2}\boldsymbol{B} & \cdots & a_{2,q}\boldsymbol{B} \\ \vdots & \vdots & & \vdots \\ a_{p,1}\boldsymbol{B} & a_{p,2}\boldsymbol{B} & \cdots & a_{p,q}\boldsymbol{B} \end{bmatrix}, \tag{8.1}$$

where $a_{i,j}$ is the element of $\boldsymbol{A}$ in i-th row and in j-th column. The dimensions (p,q,p',q') is called the configuration of the Kronecker product.

As shown in (8.1), the Kronecker product $\boldsymbol{A}\otimes\boldsymbol{B}$ is a block matrix of $p\times q$ blocks, each of which is a $p'\times q'$ sub-matrix. When $\boldsymbol{A}$ or $\boldsymbol{B}$ reduces to a scalar or a vector, the Kronecker product corresponds to some special cases. For example, when $\boldsymbol{A}$ or $\boldsymbol{B}$ is a scalar ($p=q=1$ or $p'=q'=1$), their Kronecker product $\boldsymbol{A}\otimes\boldsymbol{B}$ reduces to the scalar-matrix multiplication. When $\boldsymbol{A}$ is a column vector and $\boldsymbol{B}$ is a row vector ($q=p'=1$), the Kronecker product $\boldsymbol{A}\otimes\boldsymbol{B}$ is equivalent to their outer product, which gives a $p\times q'$ matrix. So does the case when $\boldsymbol{A}$ is a column vector and $\boldsymbol{B}$ is a row vector. One exception is that when $\boldsymbol{A}$ is a matrix and $\boldsymbol{B}$ is a column vector, the Kronecker product is not equivalent to matrix-vector multiplication (but is the same as the element-wise matrix-vector multiplication used in modern programming languages).

The arithmetic properties of the Kronecker product are listed below.

Proposition 8.1. *For $c\in\mathbb{R}$, $\boldsymbol{A}_1,\boldsymbol{A}_2\in\mathbb{R}^{p\times q}$, $\boldsymbol{B}_1,\boldsymbol{B}_2\in\mathbb{R}^{p'\times q'}$ and $\boldsymbol{C}\in\mathbb{R}^{p''\times q''}$, we have*

(i) $c(\boldsymbol{A}_1\otimes\boldsymbol{B}_1)=(c\boldsymbol{A}_1)\otimes\boldsymbol{B}_1=\boldsymbol{A}_1\otimes(c\boldsymbol{B}_1)$;

(ii) $(\boldsymbol{A}_1+\boldsymbol{A}_2)\otimes\boldsymbol{B}_1=\boldsymbol{A}_1\otimes\boldsymbol{B}_1+\boldsymbol{A}_2\otimes\boldsymbol{B}_1$ *and* $\boldsymbol{A}_1\otimes(\boldsymbol{B}_1+\boldsymbol{B}_2)=\boldsymbol{A}_1\otimes\boldsymbol{B}_1+\boldsymbol{A}_1\otimes\boldsymbol{B}_1$;

(iii) $(\boldsymbol{A}_1\otimes\boldsymbol{B}_1)\otimes\boldsymbol{C}=\boldsymbol{A}_1\otimes(\boldsymbol{B}_1\otimes\boldsymbol{C})$;

(iv) If $pq=1$ *or* $p'q'=1$ *or* $pq'=1$ *or* $qp'=1$, *then* $\boldsymbol{A}_1\otimes\boldsymbol{B}_1=\boldsymbol{B}_1\otimes\boldsymbol{A}_1$.

In summary, Proposition 8.1 shows that the Kronecker product is a bilinear operator with the associative law and the distributive law. However, in general, the Kronecker product is not commutative except for the special cases when the product is reduced to scalar-matrix multiplication or vector outer product as shown in item (iv) in Proposition 8.1.

The properties in Proposition 8.1 can be viewed as immediate results from the property of tensor product. The tensor product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is a four-way tensor such that

$$[\boldsymbol{A}\otimes_T\boldsymbol{B}]_{i,j,k,l}=[\boldsymbol{A}]_{i,j}[\boldsymbol{B}]_{k,l},$$

where $\otimes_T$ denotes the tensor product and matrices are treated as two-way tensors. The Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ is a matricization of $\boldsymbol{A} \otimes_T \boldsymbol{B}$ by collapsing the first two dimensions to rows and collapsing the last two dimensions to columns. The properties $(i) - (iii)$ in Proposition 8.1 follows immediately. When at least two of the four dimensions of $\boldsymbol{A} \otimes_T \boldsymbol{B}$ are trivial, the four-way tensor $\boldsymbol{A} \otimes_T \boldsymbol{B}$ reduces to a two-way tensor or matrix. Under this circumstance, the Kronecker product is commutative.

8.2 Kronecker Product Decomposition

In singular value decomposition (SVD), a matrix is represented as the sum of rank one matrices, and each rank one matrix is written as the outer product of the left singular vector and its corresponding right singular vector (after the transpose). Specifically, for a $p \times q$ matrix $\boldsymbol{M}$, we have

$$\boldsymbol{M} = \sum_{k=1}^{r} \lambda_k u_k v_k', \tag{8.2}$$

where $r = \min\{p, q\}$, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r \geqslant 0$ are the singular values, and $u_k \in \mathbb{R}^p$, $v_k \in \mathbb{R}^q$ are the corresponding left and right singular vectors satisfying

$$u_k' u_l = v_k' v_l = \delta_{kl} = \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{if } k \neq l. \end{cases}$$

Similarly, the Kronecker product gives another decomposition of matrix as in Definition 8.2.

Definition 8.2. *The Kronecker Product Decomposition (KPD) of a $(pp') \times (qq')$ real matrix $\boldsymbol{M}$ is*

$$\boldsymbol{M} = \sum_{k=1}^{d} \lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k, \tag{8.3}$$

where $d = \min\{pq, p'q'\}$ is the number of terms, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r \geqslant 0$ are the KPD coefficients, and $\boldsymbol{A}_k \in \mathbb{R}^{p \times q}$, $\boldsymbol{B}_k \in \mathbb{R}^{p' \times q'}$ satisfy

$$\operatorname{tr}[\boldsymbol{A}_k \boldsymbol{A}_l'] = \operatorname{tr}[\boldsymbol{B}_k \boldsymbol{B}_l'] = \delta_{kl}, \quad \textit{for all } k, l = 1, \ldots, d. \tag{8.4}$$

In addition, (p, q, p', q') is called the configuration of the KPD.

The orthonormal condition (8.4) is similar to the one used in SVD except that the vector inner product is replaced with the trace inner product. The normalizing condition embedded in (8.4) is $\mathrm{tr}[\boldsymbol{A}_k\boldsymbol{A}_k']$, which is equivalent to $\|\boldsymbol{A}_k\|_F = 1$ such that all $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are required to have a unit Frobenius norm. When the coefficients $\lambda_1, \cdots, \lambda_k$ are distinct, the terms in the KPD (8.3) are can be uniquely determined up to a sign change of $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$.

The singular value decomposition is a special case of Kronecker product decomposition when the configuration is set to $(p, 1, 1, q')$. In this case, the orthonormal condition (8.4) is the same as the one in SVD.

Note that, the KPD defined in Definition 8.2 is configuration-related. A $P \times Q$ matrix $\boldsymbol{M}$ can be decomposed with respect to any configuration (p, q, p', q') such that p is a factor of P , q is a factor of Q and $p' = P/p$, $q' = Q/q$. Therefore, there are usually multiple ways to decompose $\boldsymbol{M}$ in the form of (8.3), corresponding to different configurations. As discussed above, for a $P \times Q$ matrix $\boldsymbol{M}$, two of the possible configurations, $(P, 1, 1, Q)$ and $(1, Q, P, 1)$, are equivalent to the SVD of $\boldsymbol{M}$. In addition, two configurations, $(1, 1, P, Q)$ and $(P, Q, 1, 1)$, are the trivial cases where $d = 1$ and the only Kronecker product is a scalar-matrix multiplication.

8.3 The Rearrangement Operator

Although the SVD can be viewed as a special case of KPD, a general Kronecker product with any configuration is closely related to the vector outer product as well, as pointed out by Van Loan and Pitsianis (1993).

Denote by $\mathrm{vec}(\cdot)$ the vectorization of a matrix by stacking its rows. If $\boldsymbol{A} = (a_{ij})$ is a $p \times q$ matrix, then

$$\mathrm{vec}(\boldsymbol{A}) := [a_{1,1}, \ldots, a_{1,q}, \ldots, a_{q,1}, \ldots, q_{p,q}]'.$$

If $\boldsymbol{B} = (b_{ij})$ is a $p' \times q'$ matrix, then $\mathrm{vec}(\boldsymbol{A})[\mathrm{vec}(\boldsymbol{B})]'$ is a $(pq) \times (p'q')$ matrix containing the same set of elements as the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$, but in different positions. We define the rearrangement operator $\mathcal{R}$ in Definition 8.3 to represent this relationship.

Definition 8.3. *For $\boldsymbol{M} \in \mathbb{R}^{P\times Q}$, given a configuration (p,q,p',q'), the rearrangement operator $\mathcal{R}_{p,q} : \mathbb{R}^{P\times Q} \to \mathbb{R}^{pq\times PQ/pq}$ is defined as*

$$\mathcal{R}_{p,q}[\boldsymbol{M}] = \left[\operatorname{vec}\left(\boldsymbol{M}_{1,1}^{p',q'}\right), \ldots, \operatorname{vec}\left(\boldsymbol{M}_{1,q}^{p',q'}\right), \ldots, \operatorname{vec}\left(\boldsymbol{M}_{p,1}^{p',q'}\right), \ldots, \operatorname{vec}\left(\boldsymbol{M}_{p,q}^{p',q'}\right)\right]', \tag{8.5}$$

where $\boldsymbol{M}_{i,j}^{p',q'}$ denotes the (i,j)-th block of size $p'\times q'$ in $\boldsymbol{M}$.

The rearrangement operator $\mathcal{R}_{p,q}$ is configuration related. The subscript (p,q) emphasizes such a dependence. When there is no ambiguity, we may omit the subscript and simply use $\mathcal{R}$ for simplicity in the rest of this thesis. Some properties of the rearrangement operator are provided in Proposition 8.2

Proposition 8.2. *Let $c_1, c_2 \in \mathbb{R}$, $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathbb{R}^{P\times Q}$, $\boldsymbol{A} \in \mathbb{R}^{p\times q}$, $\boldsymbol{B} \in \mathbb{R}^{p'\times q'}$. Then*

(i) $\mathcal{R}_{p,q}[\boldsymbol{A}\otimes\boldsymbol{B}] = \operatorname{vec}(\boldsymbol{A})[\operatorname{vec}(\boldsymbol{B})]'$;

(ii) $\mathcal{R}_{p,q}[c_1\boldsymbol{M}_1 + c_2\boldsymbol{M}_2] = c_1\mathcal{R}_{p,q}[\boldsymbol{M}_1] + c_2\mathcal{R}_{p,q}[\boldsymbol{M}_2]$;

(iii) $\mathcal{R}_{p,q}^{-1}[\mathcal{R}_{p,q}[\boldsymbol{M}_1]] = \boldsymbol{M}_1$;

(iv) $\|\mathcal{R}_{p,q}[\boldsymbol{M}_1]\|_F = \|\boldsymbol{M}_1\|_F$;

(v) $\mathcal{R}_{p,q}$ *is an isomorphism;*

(vi) $\mathcal{R}_{p,q}$ *is isometric under Frobenius norm.*

Item (i) in Proposition 8.2 gives the correspondence between the Kronecker product and the outer product of two vectors. (ii) and (iii) shows the rearrangement operator is linear and bijective, resulting an isomorphism in (v). Since $\mathcal{R}$ only changes the order and shape of a matrix, the Frobenius norm is preserved as in (iv), which gives that $\mathcal{R}$ is isometric in (vi).

Not only does item (i) in Proposition 8.2 establish a connection between one Kronecker product and one vector outer product, but it also connects a general KPD in (8.3) to the form of singular value decomposition. To see this, we apply the rearrangement operator $\mathcal{R}_{p,q}$ on a Kronecker product decomposition with configuration (p,q,p',q') as in (8.3) such

that

$$\mathcal{R}_{p,q}\left[\sum_{k=1}^{d}\lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k\right] = \sum_{k=1}^{d}\lambda_k \mathcal{R}_{p,q}[\boldsymbol{A}_k \otimes \boldsymbol{B}_k] = \sum_{k=1}^{d}\lambda_k \mathrm{vec}(\boldsymbol{A}_k)[\mathrm{vec}(\boldsymbol{B}_k)]'. \tag{8.6}$$

The right hand side of (8.6) is exactly a singular value decomposition.

CHAPTER 9

KoPA: Automatic Kronecker Product Approximation

9.1 Introduction

Observations that are matrix/tensor valued have been commonly seen in various scientific fields and social studies. In recent years, technological advances have made high dimensional matrix/tensor type data that are possible and more and more prevalent. Examples include high resolution images in face recognition and motion detection (Turk and Pentland, 1991; Bruce and Young, 1986; Parkhi et al., 2015), brain images through fMRI (Belliveau et al., 1991; Maldjian et al., 2003), adjacent matrices of social networks of millions of nodes (Goldenberg et al., 2010), the covariance matrix of thousands of stock returns (Ng et al., 1992; Fan et al., 2011), the import/export network among hundreds of countries (Chen et al., 2019a), etc. Due to the high dimensionality of the data, it is often useful and preferred to store, compress, represent, or summarize the matrices/tensors through low dimensional structures. In particular, low rank approximations of matrices have been ubiquitous. Finding a low rank approximation of a given matrix is closely related to the singular value decomposition (SVD), see Eckart and Young (1936) for an early paper pointing out the connection. SVD has proven to be extremely useful in matrix completion (Candès and Recht, 2009; Candes and Plan, 2010; Cai et al., 2010), community detection (Le et al., 2016), image denoising (Guo et al., 2015), among many others.

In this chapter, we focus on the model

$$\boldsymbol{Y} = \lambda \boldsymbol{A} \otimes \boldsymbol{B} + \sigma \boldsymbol{E},$$

where $\boldsymbol{E}$ is a standard Gaussian ensemble consisting of IID standard normal entries, $\lambda > 0$ and $\sigma > 0$ indicate the strength of signal and noise respectively. We consider the matrix de-noising problem which aims to recover the Kronecker product $\lambda \boldsymbol{A} \otimes \boldsymbol{B}$ from the noisy observation $\boldsymbol{Y}$.

9.2 Framework

9.2.1 Kronecker Product Model

We consider the model where the observed $P \times Q$ matrix $\boldsymbol{Y}$ is a noisy version of an unknown Kronecker product

$$\boldsymbol{Y} = \lambda \boldsymbol{A} \otimes \boldsymbol{B} + \frac{\sigma}{\sqrt{PQ}} \boldsymbol{E}. \tag{9.1}$$

To resolve the obvious unidentifiability regarding $\boldsymbol{A}$ and $\boldsymbol{B}$, we require

$$\|\boldsymbol{A}\|_F = \|\boldsymbol{B}\|_F = 1, \tag{9.2}$$

so that $\lambda > 0$ indicates the strength of the signal part. Note that under (9.2), $\boldsymbol{A}$ and $\boldsymbol{B}$ are identified up to a sign change. We assume that the noise matrix $\boldsymbol{E}$ has IID stand normal entries, and consequently the strength of the noise is controlled by $\sigma > 0$. The dimensions of $\boldsymbol{A}$ and $\boldsymbol{B}$ correspond to the integer factorization of the dimension of $\boldsymbol{Y}$. For convenience, we assume throughout this article that the dimension of the observed matrix $\boldsymbol{Y}$ in (9.1) is $2^M \times 2^N$ with $M, N \in \mathbb{N}$. As a result, the dimension of $\boldsymbol{A}$ must be of the form $2^{m_0} \times 2^{n_0}$, where $0 \leqslant m_0 \leqslant M$ and $0 \leqslant n_0 \leqslant N$, and the corresponding dimension of $\boldsymbol{B}$ is $2^{m_0^\dagger} \times 2^{n_0^\dagger}$, where $m_0^\dagger = M - m_0$ and $n_0^\dagger = N - n_0$. Therefore, we can simply use the pair (m_0, n_0) to denote the configuration of the Kronecker product in (9.1). An implicit advantage of this assumption lies in the fact that if two configurations (m, n) and (m', n') are different, then the number of rows of $\boldsymbol{A}$ under one configurations divides the one under the other, and

similarly for the numbers of columns, and for $\boldsymbol{B}$. For example, if $m \leqslant m'$, then the number of rows of $\boldsymbol{A}$ under the former configuration, which is 2^m, divides the number of rows $2^{m'}$ under the latter one. This fact leads to a more elegant treatment of the theoretical analysis in Section 9.3.

For image analysis, assuming the dimension to be powers of 2 seems rather reasonable. On the other hand, for other applications where the dimension of the observed matrix are not powers of 2, one can transform the matrix to fulfill the assumption. For example, one can super-sample the matrix to increase the dimension to the closest powers of 2, or augment the matrix by padding zeros. The methodology proposed in this paper can be applied to any integer numbers P and Q with more than two factors.

We will consider two mechanisms for the signal part $\lambda \boldsymbol{A} \otimes \boldsymbol{B}$.

Deterministic Scheme. We assume that λ, $\boldsymbol{A}$ and $\boldsymbol{B}$ are deterministic, satisfying (9.2). We define the following signal-to-noise ratio to measure the signal strength

$$\frac{\|\lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2}{\mathbb{E}\|\sigma \boldsymbol{E}/2^{(M+N)/2}\|_F^2} = \frac{\lambda^2}{\sigma^2}.$$

Random Scheme. Assume that λ, $\boldsymbol{A}$ and $\boldsymbol{B}$ are random and independent with $\boldsymbol{E}$. Although $\boldsymbol{A}$ and $\boldsymbol{B}$ are stochastic, we assume that they have been rescaled so that (9.2) is fulfilled. In this case the signal-to-noise ratio is defined as

$$\frac{\mathbb{E}\|\lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2}{\mathbb{E}\|\sigma \boldsymbol{E}/2^{(M+N)/2}\|_F^2} = \frac{\mathbb{E}\,\lambda^2}{\sigma^2}.$$

We distinguish between these two schemes to account for the different assumptions on data generating mechanism. In the random scheme, the observed matrix data is assumed to be randomly chosen from a (super-)population of matrices with an ad-hoc prior, which is Kronecker product of two independent Gaussian random matrices here. Under the random scheme assumption, ill-behaved matrices arise with negligible probabilities under the prior. Similar assumptions have been used in factor analysis and random effects models. The deterministic scheme incorporates arbitrary matrices. Additional assumptions need to be imposed to exclude extreme cases for which the proposed model selection would fail.

9.2.2 Estimation with a Known Configuration

Suppose we want to estimate $\boldsymbol{A}$ and $\boldsymbol{B}$ based on a given configuration (m,n), that is, the dimensions of $\boldsymbol{A}$ and $\boldsymbol{B}$ are $2^m \times 2^n$ and $2^{m^\dagger} \times 2^{n^\dagger}$ respectively. Again we use $m^\dagger = M - m$ and $n^\dagger = N - n$ to ease the notation when M and N are known. To estimate $\boldsymbol{A}$ and $\boldsymbol{B}$ in (9.1) from the observed matrix $\boldsymbol{Y}$, we solve the minimization problem

$$\min_{\lambda,\boldsymbol{A},\boldsymbol{B}} \|\boldsymbol{Y} - \lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2, \quad \text{subject to } \|\boldsymbol{A}\|_F = \|\boldsymbol{B}\|_F = 1. \tag{9.3}$$

Since we have assumed that the noise matrix contains IID standard normal entries, (9.3) is also equivalent to the MLE. This optimization problem has been formulated as the nearest Kronecker product (NKP) problem in the matrix computation literature (Van Loan and Pitsianis, 1993), and solved through the SVD after rearrangement. According to Section 8.2, after applying the rearrangement operator, the cost function in (9.3) is equivalent to

$$\|\boldsymbol{Y} - \lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2 = \|\mathcal{R}[\boldsymbol{Y}] - \lambda \mathrm{vec}(\boldsymbol{A})[\mathrm{vec}(\boldsymbol{B})]'\|_F^2.$$

We note that the rearrangement operator $\mathcal{R}$ defined in (8.5) depends on the configuration of the block matrix, and in the current case, on the configuration (m,n). Let $\mathcal{R}[\boldsymbol{Y}] = \sum_{k=1}^{d} \lambda_k u_k v_k'$ be the SVD of the rearranged matrix $\mathcal{R}_{m,n}[\boldsymbol{Y}]$, where $\lambda_1 \geqslant \cdots \geqslant \lambda_d \geqslant 0$ are the singular values in decreasing order, u_k and v_k are the corresponding left and right singular vectors and $d = 2^{m+n} \wedge 2^{m^\dagger + n^\dagger}$. The estimators for model (9.1) are given by

$$\hat{\lambda} = \lambda_1 = \|\mathcal{R}[\boldsymbol{Y}]\|_S, \quad \hat{\boldsymbol{A}} = \mathrm{vec}^{-1}(u_1), \quad \hat{\boldsymbol{B}} = \mathrm{vec}^{-1}(v_1), \quad \hat{\sigma}^2 = \|\boldsymbol{Y}\|_F^2 - \hat{\lambda}^2, \tag{9.4}$$

where vec^{-1} is the inverse operation of $\mathrm{vec}(\cdot)$ that restores a vector back into a matrix of proper dimensions.

We exam a few special cases of the configuration (m,n). When $(m,n) = (0,0)$ or $(m,n) = (M,N)$, the nearest Kronecker product approximation of $\boldsymbol{Y}$ is always itself. For instance,

if $m = n = 0$, the estimators are

$$\hat{\lambda} = \|\boldsymbol{Y}\|_F, \quad \hat{\boldsymbol{A}} = 1, \quad \hat{\boldsymbol{B}} = \hat{\lambda}^{-1}\boldsymbol{Y}, \quad \hat{\sigma}^2 = 0.$$

These two configurations are obviously over-fitting, and we shall exclude them in the subsequent analysis.

When $(m,n) = (0,N)$ or $(m,n) = (M,0)$, the nearest Kronecker product approximation of $\boldsymbol{Y}$ is the same as the rank-1 approximation of $\boldsymbol{Y}$ without rearrangement. When the true configuration used to generate $\boldsymbol{Y}$ is chosen, that is $(m,n) = (m_0,n_0)$, the problem is equivalent to denoising a perturbed rank-1 matrix, since

$$\mathcal{R}_{m_0,n_0}[\boldsymbol{Y}] = \lambda \mathrm{vec}(\boldsymbol{A})\mathrm{vec}(\boldsymbol{B})' + \frac{\sigma}{2^{(M+N)/2}}\mathcal{R}_{m_0,n_0}[\boldsymbol{E}], \tag{9.5}$$

where the rearranged noise matrix $\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]$ is still a standard Gaussian ensemble. Therefore λ, $\boldsymbol{A}$ and $\boldsymbol{B}$ can be recovered consistently when $\sigma\|\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]\|_S = o_p(\lambda 2^{(M+N)/2})$. Details will be discussed in Section 9.3.

9.2.3 Configuration Determination through an Information Criteria

Our primary goal is to recover the Kronecker product $\lambda\boldsymbol{A}\otimes\boldsymbol{B}$ from $\boldsymbol{Y}$, based on model (9.1). It depends on the configuration of the Kronecker product, which is typically unknown. We propose to use the information criterion based procedure to select the configuration.

Recall that the dimension of $\boldsymbol{Y}$ is $2^M \times 2^N$. If the dimension of $\boldsymbol{A}$ is $2^m \times 2^n$, then the dimension of $\boldsymbol{B}$ must be $2^{m^\dagger} \times 2^{n^\dagger}$, where $m^\dagger = M - m$ and $n^\dagger = N - n$. Therefore, the configuration can be indexed by the pair (m,n), which takes value from the Cartesian product set $\{0,\ldots,M\}\times\{0,\ldots,N\}$.

For any given configuration (m,n), the estimation procedure in Section 9.2.2 leads to the corresponding estimators $\hat{\lambda}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$. Denote the estimated Kronecker product by $\hat{\boldsymbol{Y}}^{(m,n)} = \hat{\lambda}\hat{\boldsymbol{A}}\otimes\hat{\boldsymbol{B}}$. Note that all of $\hat{\lambda}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ depend implicitly on the configuration (m,n) used in estimation, and should be written as $\hat{\lambda} = \hat{\lambda}^{(m,n)}$ etc. However, we will suppress the configuration index from the notation for simplicity, whenever its meaning is clear in the

context. Under the assumption that the noise matrix $\boldsymbol{E}$ is a standard Gaussian ensemble, we define the information criterion as

$$\mathrm{IC}_\kappa(m,n) = 2^{M+N} \ln \|\boldsymbol{Y} - \hat{\boldsymbol{Y}}^{(m,n)}\|_F^2 + \kappa\eta, \tag{9.6}$$

where $\eta = 2^{m+n} + 2^{m^\dagger+n^\dagger}$ is the number of parameters involved in the Kronecker product of the configuration (m,n), and $\kappa \geqslant 0$ controls the penalty on the model complexity. The information criterion (9.6) can be viewed as an extended version of the BIC. Similar proposals have been introduced by Chen and Chen (2008) and Foygel and Drton (2010) in the linear regression and graphical models setting, respectively. The information criterion (9.6) reduces to the log mean square error when $\kappa = 0$, and corresponds to the Akaike information criterion (AIC) (Akaike, 1998) when $\kappa = 2$, and the Bayesian information criterion (BIC) (Schwarz, 1978) when $\kappa = \ln 2^{M+N} = (M+N)\ln 2$.

Strictly speaking, the number of parameters involved in the Kronecker product $\lambda\boldsymbol{A}\otimes\boldsymbol{B}$ should be $2^{m+n} + 2^{m^\dagger+n^\dagger} - 1$ because of the constraints (9.2). Since it does not affect the selection procedure to be introduced in (9.7), we will use $\eta = 2^{m+n} + 2^{m^\dagger+n^\dagger}$ for simplicity.

The information criterion (9.6) can be calculated for all configurations, and the one corresponding to the smallest value of (9.6) will be selected, based on which the estimation procedure in Section 9.2.2 proceeds. In other words, the selected configuration $(\hat{m},\hat{n})$ is obtained through

$$(\hat{m},\hat{n}) = \underset{(m,n)\in\mathcal{C}}{\arg\min}\ \mathrm{IC}_\kappa(m,n), \tag{9.7}$$

where $\mathcal{C}$ is the set of all candidate configurations.

As discussed in Section 9.2.2, when $m = n = 0$ or $(m,n) = (M,N)$, it holds that $\hat{\boldsymbol{Y}} = \boldsymbol{Y}$, and the information criterion (9.6) will be $-\infty$, no matter what value κ takes. Therefore, these two configurations should be excluded in model selection and we use

$$\mathcal{C} := \{0,\ldots,M\}\times\{0,\ldots,N\}\setminus\{(0,0),(M,N)\},$$

as the set of candidate configurations in (9.7). Note that the set $\{0,\ldots,M\}\times\{0,\ldots,N\}$ forms a rectangle lattice in $\mathbb{Z}^2$, and $(m,n) = (0,0)$ and $(m,n) = (M,N)$ are the bottom left

and top right corner of the lattice. Therefore, we sometimes intuitively refer to these two configurations as the "corner cases" in the sequel. Furthermore, define $\mathcal{W}$ as the set of all wrong configurations

$$\mathcal{W} := \mathcal{C} \setminus \{(m_0, n_0)\}.$$

We now provide a heuristic argument to show how the selection procedure (9.7) is able to select the true configuration (m_0, n_0). We will leave some technical results aside, and only highlight the essential idea. Precise statements and their rigorous analysis will be presented in Section 9.3. For simplicity, assume that λ, σ and κ are fixed constants. Also assume that both $(m_0 + n_0)$ and $(m_0^\dagger + n_0^\dagger)$ diverge, so that the number of parameters $\eta_0 = 2^{m_0+n_0} + 2^{m_0^\dagger+n_0^\dagger}$ is of a smaller magnitude than 2^{M+N}.

According to (9.4), for a given configuration (m, n), $\mathcal{R}_{m,n}[\hat{\boldsymbol{Y}}]$ equals the first SVD component of $\mathcal{R}_{m,n}[\boldsymbol{Y}]$, and it follows that $\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|_F^2 = \|\boldsymbol{Y}\|_F^2 - \|\hat{\boldsymbol{Y}}\|_F^2 = \|\boldsymbol{Y}\|_F^2 - \hat{\lambda}^2$, and the information criterion (9.6) can be rewritten as

$$\mathrm{IC}_\kappa(m, n) = 2^{M+N} \ln(\|\boldsymbol{Y}\|_F^2 - \hat{\lambda}^2) + \kappa\eta. \tag{9.8}$$

For the true configuration $(m, n) = (m_0, n_0)$, the rearranged matrix $\mathcal{R}_{m_0,n_0}[\boldsymbol{Y}]$ takes the form (9.5), where the first term is a rank-1 matrix of spectral norm λ, and the noise term has a spectral norm of the order $O(2^{-(m_0+n_0)/2} + 2^{-(m_0^\dagger+n_0^\dagger)/2})$ (details given in Section 9.3), which is negligible relative to λ, under the assumption $m_0 + n_0 \gg 1, m_0^\dagger + n_0^\dagger \gg 1$. So under the true configuration, $\hat{\lambda} \approx \lambda$. On the other hand, the number of parameters $\eta_0 = o(2^{M+N})$, making the penalty term much smaller than the log likelihood in (9.6). To summarize,

$$\mathrm{IC}_\kappa(m_0, n_0) \approx 2^{M+N} \ln\left[\|\lambda \boldsymbol{A} \otimes \boldsymbol{B} + \sigma\, 2^{-(M+N)/2} \boldsymbol{E}\|_F^2 - \lambda^2\right] \approx 2^{M+N} \ln \sigma^2.$$

For a wrong configuration $(m, n) \in \mathcal{W}$ that is close to the true one, the spectrum norm $\|\mathcal{R}_{m,n}[\boldsymbol{E}]\|_S$ and the number of parameters η are still negligible. However, the estimated coefficient $\hat{\lambda}$ is smaller than λ since

$$\hat{\lambda} = \|\mathcal{R}_{m,n}[\boldsymbol{Y}]\|_S \approx \|\mathcal{R}_{m,n}[\lambda \boldsymbol{A} \otimes \boldsymbol{B}]\|_S < \lambda.$$

Let us assume that $\|\mathcal{R}_{m,n}[\lambda \boldsymbol{A} \otimes \boldsymbol{B}]\|_S \leq \phi\lambda$ for some $0 < \phi < 1$, which implies that for the wrong configuration (m, n),

$$\begin{aligned}
\mathrm{IC}_\kappa(m,n) &\approx 2^{M+N} \ln\left[\|\lambda \boldsymbol{A} \otimes \boldsymbol{B} + \sigma 2^{-(M+N)/2} \boldsymbol{E}\|_F^2 - \hat{\lambda}^2\right] \\
&\approx 2^{M+N} \ln\left[\|\sigma 2^{-(M+N)/2} \boldsymbol{E}\|_F^2 + \lambda^2 - \phi^2\lambda^2\right] \\
&\approx 2^{M+N} \ln\left[\sigma^2\left(1 + \frac{(1-\phi^2)\lambda^2}{\sigma^2}\right)\right].
\end{aligned}$$

Therefore, the information criterion (9.6) is in favor of the true configuration over a wrong but close-to-truth one, and the two quantities are separated by

$$\mathrm{IC}_\kappa(m,n) - \mathrm{IC}_\kappa(m_0,n_0) \approx 2^{M+N} \ln[1 + (1-\phi^2)\lambda^2/\sigma^2].$$

On the other hand, for a wrong configuration $(m, n) \in \mathcal{W}$ that is close to the corner configuration $(0, 0)$ or (M, N), the singular value $\|\mathcal{R}_{m,n}[\boldsymbol{E}]\|_S$ can be as large as $1/2$, making the separation between $\mathrm{IC}_\kappa(m, n)$ and $\mathrm{IC}_\kappa(m_0, n_0)$ by the log likelihood not guaranteed, i.e. it can happen that $\hat{\lambda} > \lambda$ under the wrong configuration. But at the same time the number of parameters η is also approximately 2^{M+N} , so $\mathrm{IC}_\kappa(m, n)$ receives a heavy penalty, which once again makes it greater than $\mathrm{IC}_\kappa(m_0, n_0)$.

In summary, the trade-off between log likelihood and model complexity plays its role here, as expected. Wrong but close-to-truth configurations involve similar numbers of parameters as the true one, but lead to much smaller likelihoods. On the other hand, a close-to-corner configuration may yield a $\hat{\boldsymbol{Y}}$ closer to the original $\boldsymbol{Y}$, but requires much more parameters to do so. The true configuration can thus be selected because it reaches the optimal balance between the the likelihood and model complexity.

In the preceding discussion we have assumed many convenient conditions to simplify the arguments and signify the essential idea. In particular, by assuming that λ is a positive constant, the signal strength in model (9.1) is quite strong. In Section 9.3 we will make effort to establish the model selection consistency under minimal conditions.

9.2.4 Multi-term Kronecker Product Models

In this section, we extend the one-term Kronecker product model in (9.1) to the following K-term Kronecker product model.

$$\boldsymbol{Y} = \sum_{k=1}^{K} \lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k + \frac{\sigma}{2^{(M+N)/2}} \boldsymbol{E}, \tag{9.9}$$

where $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_K > 0$ and $\boldsymbol{A}_k \in \mathbb{R}^{2^{m_0} \times 2^{n_0}}$, $\boldsymbol{B}_k \in \mathbb{R}^{2^{m_0^\dagger} \times 2^{n_0^\dagger}}$, $k = 1, \cdots, K$ satisfy the following orthonormal condition:

$$\mathrm{tr}(\boldsymbol{A}_k \boldsymbol{A}_l') = \mathrm{tr}(\boldsymbol{B}_k \boldsymbol{B}_l') = \delta_{kl} := \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{if } k \neq l. \end{cases}$$

The orthonormal condition implies the identifiability: if $\lambda_1 > \lambda_2 > \cdots > \lambda_k > 0$, then $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are identified up to a sign change, see Section 8.2. Note that the K terms in model (9.9) have the same configuration (m_0, n_0). Therefore, although multiple terms are present, there is only one configuration to be determined.

Once the configuration is given, the rearranged $\boldsymbol{Y}$ becomes the sum of a rank K matrix and a noise matrix. The determination of K turns into the rank selection problem in low rank approximation, and existing methods (Bai, 2003; Ahn and Horenstein, 2013) can be applied. Therefore, we focus on the choice of the configuration for model (9.9). We propose to use the same procedure based on the one-term model, although there are actually K terms in model (9.9). We show that, if the leading term in (9.9) is strong enough, i.e. if λ_1 is large enough, compared with other λ_k as well as σ, the information criterion introduced in Section 9.2.3 will continue to select the true configuration consistently.

9.3 Theoretical Analysis

In this section we provide a theoretical guarantee of the configuration selection procedure proposed in Section 9.2.3, by establishing its asymptotic consistency. Throughout this section all our discussion will be based on model (9.1).

9.3.1 Assumptions and Estimation Consistency under Known Configuration

We first introduce the assumptions of the theoretical analysis. Recall that for model (9.1), (m_0, n_0) denotes the true configuration, i.e. the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are of dimensions $2^{m_0} \times 2^{n_0}$ and $2^{m_0^\dagger} \times 2^{n_0^\dagger}$ respectively. For the asymptotic analysis, we make the following assumption on the sizes of $\boldsymbol{A}$ and $\boldsymbol{B}$, which follows the paradigm of high dimensional analysis.

Assumption 9.1 (Assumption on Dimension)**.** *Consider model* (9.1)*. As* $M + N \to \infty$*, assume that the true configuration* (m_0, n_0) *satisfies*

$$\frac{m_0 + n_0}{\ln\ln(MN)} \to \infty, \quad \frac{m_0^\dagger + n_0^\dagger}{\ln\ln(MN)} \to \infty,$$

where $m_0^\dagger = M - m_0$ *and* $n_0^\dagger = N - n_0$*.*

On the one hand, the condition entails that the numbers of entries in $\boldsymbol{A}$ and $\boldsymbol{B}$ will need to diverge to infinity, and so is that of $\boldsymbol{Y}$. On the other hand, it is also ensured that the true configuration cannot stay too close to the corner configurations. On the other hand, we remark that this will be the only condition on the sizes of the involved matrices. In particular, we do not require all of $m_0, n_0, m_0^\dagger, n_0^\dagger$ to go to infinity. Consequently, the low rank approximation (when $(m_0, n_0) = (M, 0)$ or $(m_0, n_0) = (0, N)$) is also included in the KoPA framework as a special case.

The number of parameters involved in the Kronecker product $\lambda \boldsymbol{A} \otimes \boldsymbol{B}$ is $\eta_0 = 2^{m_0+n_0} + 2^{m_0^\dagger + n_0^\dagger}$. It is a much smaller number than $2^M \times 2^N$, the number of elements in $\boldsymbol{Y}$. Hence Assumption 9.1 implies a significant dimension reduction.

We also make the following assumption on the error matrix $\boldsymbol{E}$.

Assumption 9.2 (Assumption on Noise)**.** *Consider model* (9.1)*. Assume that* $\boldsymbol{E}$ *is a standard Gaussian ensemble, i.e. with IID standard normal entries.*

We conclude this subsection with the convergence rates of the estimators $\hat{\lambda}$,$\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$, given by the estimation procedure in Section 9.2.2 under the true configuration. Since

the error matrix $\boldsymbol{E}$ has IID standard normal entries, according to Vershynin (2010), the expectation of the largest singular value of the rearranged error matrix $\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]$ is bounded by

$$s_0 = 2^{(m_0+n_0)/2} + 2^{(m_0^\dagger+n_0^\dagger)/2}.$$

Theorem 9.1. *Let $\hat{\lambda}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ be the estimators obtained under the true configuration, as given in (9.4). Suppose Assumptions 9.1 and 9.2 hold, then for the deterministic scheme of model (9.1), we have*

$$\frac{\hat{\lambda}-\lambda}{\lambda} = O_p\left(\frac{r_0}{\lambda/\sigma}\right), \quad \|\hat{\boldsymbol{A}}-\boldsymbol{A}\|_F^2 = O_p\left(\frac{r_0}{\lambda/\sigma}\right), \quad \|\hat{\boldsymbol{B}}-\boldsymbol{B}\|_F^2 = O_p\left(\frac{r_0}{\lambda/\sigma}\right),$$

where

$$r_0 = \frac{s_0}{2^{(M+N)/2}} = 2^{-(m_0+n_0)/2} + 2^{-(m_0^\dagger+n_0^\dagger)/2}.$$

9.3.2 Consistency of Configuration Selection

To study the consistency of the configuration selection proposed in Section 9.2.3, we need assumptions on the signal-to-noise ratio. We choose to present model (9.1) with both λ and σ so that it is able to account for any actual data generating mechanism. On the other hand, the mathematical properties would only depend on the ratio λ/σ. The strength of the signal also depends on the contrast between true and wrong configurations. If a configuration $(m,n) \in \mathcal{W}$ is used for the estimation, $\boldsymbol{Y}$ is rearranged as

$$\mathcal{R}_{m,n}[\boldsymbol{Y}] = \lambda \mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}] + \sigma 2^{-(M+N)/2}\mathcal{R}_{m,n}[\boldsymbol{E}]. \tag{9.10}$$

Ignoring the noise term, only the first singular value component of $\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]$ (multiplied by λ) is expected to enter $\hat{\boldsymbol{Y}}$. When the true configuration is used, $\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]$ is a rank-1 matrix, and its leading singular value equals 1 (recall that we have assumed that $\|\boldsymbol{A}\|_F = \|\boldsymbol{B}\|_F = 1$). On the other hand, if a wrong configuration is used, then $\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]$ is no longer rank-1, and its leading singular value should be smaller than 1. Define

$$\phi := \max_{(m,n)\in\mathcal{W}} \|\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S.$$

The quantity ϕ characterize how much of the signal $\boldsymbol{A} \otimes \boldsymbol{B}$ can be captured by a wrong configuration, and it always holds that $0 < \phi \leq 1$, so we also introduce

$$\psi^2 := 1 - \phi^2,$$

and call it the *representation gap*. Note that $0 \leq \psi^2 < 1$, and the larger ψ^2 is, the easier it is to separate true and wrong configurations. The following assumption shows the interplay between the representation gap ψ^2 and the signal-to-noise ratio λ/σ.

Assumption 9.3 (Representation Gap)**.** *For model* (9.1)*, assume that* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are deterministic matrices, and*

$$\lim_{M+N\to\infty} \frac{2^{(M+N)/2}}{2^{(m_0+n_0)/2} + 2^{(m_0^\dagger+n_0^\dagger)/2}} \cdot (\lambda/\sigma) \cdot \psi = \infty, \tag{9.11}$$

and

$$\lim_{M+N\to\infty} 2^{(M+N)/4} \cdot (\lambda/\sigma) \cdot \psi^2 = \infty. \tag{9.12}$$

In both (9.11) and (9.12), the signal-to-noise ratio and the representation gap ψ^2 can diminish to zero, as long as they do not converge to zero too fast. In this sense, Assumption 9.3 is very flexible by requiring only very week signal strength.

We have defined ϕ as the maximum over $\mathcal{W}$, the set of all wrong configurations. In fact, if we let $\phi_{m,n} := \|\mathcal{R}_{m,n}[\boldsymbol{A} \otimes \boldsymbol{B}]\|_S$, and $\psi_{m,n}^2 = 1 - \phi_{m,n}^2$, then Assumption 9.3 can also be given through $\psi_{m,n}^2$ instead of an uniform lower bound ψ^2, leading to a weaker version of the assumption. On the other hand, as will be show in Section 9.3.3, if $\boldsymbol{A}$ and $\boldsymbol{B}$ are randomly generated according to the Random Scheme, then indeed all $\psi_{m,n}^2$ are larger than or around $1/2$ with an overwhelming probability. This is suggesting that using the lower bound ψ^2 in Assumption 9.3 for the deterministic scheme is still reasonable. Therefore, we do not spell out the detailed version of Assumption 9.3 using $\psi_{m,n}^2$, but present it in the current simple form.

Notions similar to the representation gap appear as key parameters in many other problems. For example, in variable selection of linear regression problems with all independent and univariance covariates, the representation gap would be the smallest absolute non-zero

coefficient in the model. In matrix rank determination problems or factor models, the representation gap would be the eigen-gap, or the smallest singular value.

The following theorem quantifies the separation of the information criterion (9.6) between the true and wrong configurations.

Theorem 9.2. *Consider model* (9.1)*, and assume Assumptions 9.1, 9.2, 9.3. If*

$$\kappa \geq 2\ln 2, \quad \textit{and} \quad \kappa = o\left(\frac{2^{M+N}\ln(1+(\lambda/\sigma)^2\psi^2)}{2^{m_0+n_0}+2^{m_0^\dagger+n_0^\dagger}}\right), \tag{9.13}$$

then

$$\min_{(m,n)\in\mathcal{W}} \mathbb{E}[\mathrm{IC}_\kappa(m,n)] - \mathbb{E}[\mathrm{IC}_\kappa(m_0,n_0)] \geq 2^{M+N}\cdot\ln[1+(\lambda/\sigma)^2\psi^2]\cdot(1+o(1)).$$

To be precise, we note that for a sequence of numbers $\{a_k\}$, the statement $a_k \geq o(1)$ is understood as $\max\{-a_k, 0\} = o(1)$. According to Assumptions 9.3, $(\lambda/\sigma)^2\psi^2 \gg 2^{-(M+N)/2}$, so Theorem 9.2 shows that the separation of the information criterion is at least of the order $2^{(M+N)/2}$.

The first condition in (9.13) ensures the penalty on number of parameters are large enough to exclude configurations close to $(0,0)$ and (M,N). The second condition in (9.13) is imposed so that the contribution from the penalty term under the true configuration is dominated by the representation gap. The exact formula of the difference in expected information criterion is given by (B.8) in Appendix.

Next theorem establishes the consistency of (9.6). We need to define the symbol $\gtrsim$: for two sequences of positive numbers $\{a_k\}$ and $\{b_k\}$, $a_k \gtrsim b_k$ is defined as $\liminf_{k\to\infty} a_k/b_k > 0$.

Theorem 9.3. *Assume the same conditions of Theorem 9.2, then*

$$P\left[\mathrm{IC}_\kappa(m_0,n_0) < \min_{(m,n)\in\mathcal{W}} \mathrm{IC}_\kappa(m,n)\right] \geqslant 1-\exp\left\{-C2^{M+N}+\ln(MN)\right\},$$

where the constant C depending on λ/σ and ψ is of order

$$C(\lambda/\sigma,\psi) \gtrsim (\alpha^{1/3}-1)\wedge\left(\frac{\alpha-\alpha^{2/3}}{1+\lambda/\sigma}\right)^2,$$

with $\alpha = 1 + (\lambda/\sigma)^2\psi^2$. *In particular, the preceding convergence rate implies the consistency, i.e.*

$$\lim_{M+N\to\infty} P\left[\mathrm{IC}_\kappa(m_0,n_0) < \min_{(m,n)\in\mathcal{W}} \mathrm{IC}_\kappa(m,n)\right] = 1. \tag{9.14}$$

In Assumption 9.3, we focus on the minimal signal-to-noise ratio and representation gap. On the other hand, if they are large such that $\liminf(\lambda/\sigma)^2\psi^2 \geq 1/2$, then the condition $\kappa \geq 2\ln 2$ can be dropped from Theorem 9.2 and Theorem 9.3, which continue to hold if we set $\kappa = 0$ in (9.6). In other words, if the signal strength and the representation gap are sufficiently large, one can simply use mean squared error to select the configuration. Specifically, it requires $\lambda^2\psi^2/\sigma^2 > 1/2$ to use $\kappa = 0$ in the information criterion.

9.3.3 Model Selection under Random Scheme

In this section we consider the consistency of the model selection under the random scheme (9.16). First of all, similar convergence rates as Theorem 9.1 can be obtained under the random scheme.

Corollary 9.1. *Assume Assumptions 9.1 and 9.2. If* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are generated according to the random scheme* (9.16)*, then the conclusion of Theorem 9.1 continue to hold.*

If a configuration $(m,n) \in \mathcal{W}$ is used, then the estimation procedure given in Section 9.2.2 rearranges $\boldsymbol{Y}$ as (9.10). In Section 9.3.2 for the deterministic scheme, we introduce ϕ as the upper bound of $\|\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S$ over all wrong configurations. For the random scheme, it turns out this upper bound and hence the representation gap ψ, depending on $\boldsymbol{A}$ and $\boldsymbol{B}$, is also random. We introduce the following "random" version of Assumption 9.3.

Assumption 9.4 (Representation Gap)**.** *Assume in model* (9.1)*,* λ*,* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are random and independent with* $\boldsymbol{E}$*. Assume there exist two sequences of positive numbers* $\{\lambda_0\}$ *and* $\{\psi_0\}$ *satisfying* (9.11) *and* (9.12) *(by replacing* λ *and* ψ *therein), such that*

$$\limsup_{M+N\to\infty} \mathbb{E}[\lambda^2/\lambda_0^2] < \infty, \quad \limsup_{M+N\to\infty} \mathbb{E}[\psi^2/\psi_0^2] < \infty,$$

and for any constant $c > 0$

$$\lim_{M+N\to\infty} MN \cdot P\left[\lambda^2/\lambda_0^2 < 1-c\right] = \lim_{M+N\to\infty} MN \cdot P\left[\psi^2/\psi_0^2 < 1-c\right] = 0. \tag{9.15}$$

With Assumption 9.4, Theorem 9.2 and 9.3 continue to hold under the random scheme, as asserted by the next theorem.

Theorem 9.4. *Consider model* (9.1) *with random* λ, $\boldsymbol{A}$ *and* $\boldsymbol{B}$. *Under Assumptions 9.1, 9.2 and 9.4, it holds that*

$$\min_{(m,n)\in\mathcal{W}} \mathbb{E}[\mathrm{IC}_\kappa(m,n)] - \mathbb{E}[\mathrm{IC}_\kappa(m_0,n_0)] \geq 2^{M+N} \cdot \ln[1+(\lambda_0/\sigma)^2\psi_0^2] \cdot (1+o(1)).$$

Furthermore, the consistency (9.14) *holds.*

Assumption 9.4 is formulated to single out the minimal condition required for the consistency under the random scheme. There is no specific distribution assumptions imposed on $\boldsymbol{A}$ and $\boldsymbol{B}$. In the rest of this section, we demonstrate that how it can be satisfied under normality.

Example 9.1. *Consider model* (9.1). *Suppose that*

$$\lambda \boldsymbol{A}\otimes\boldsymbol{B} = \frac{\lambda_0 \tilde{\boldsymbol{A}}\otimes\tilde{\boldsymbol{B}}}{2^{(M+N)/2}}, \tag{9.16}$$

where λ_0 *is deterministic, and* $\tilde{\boldsymbol{A}}$ *and* $\tilde{\boldsymbol{B}}$ *are independent, and both consisting of IID standard normal entries. In order to fulfill the identifiability condition* (9.2), *we let* $\boldsymbol{A} = \tilde{\boldsymbol{A}}/\|\tilde{\boldsymbol{A}}\|_F$, $\boldsymbol{B} = \tilde{\boldsymbol{B}}/\|\tilde{\boldsymbol{B}}\|_F$, *and* $\lambda = \lambda_0 \cdot \|\tilde{\boldsymbol{A}}\|_F \cdot \|\tilde{\boldsymbol{B}}\|_F/2^{(M+N)/2}$. *Also assume that* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are both independent with* $\boldsymbol{E}$. *For this example, the signal-to-noise ratio becomes*

$$\frac{\mathbb{E}\|\lambda\boldsymbol{A}\otimes\boldsymbol{B}\|_F^2}{\mathbb{E}\|\sigma\boldsymbol{E}/2^{(M+N)/2}\|_F^2} = \frac{\lambda_0^2}{\sigma^2}.$$

Recall that ϕ is defined as the upper bound of $\|\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S$ over all wrong configurations. Only when the true configurations (m_0,n_0) is used, the rearrangement $\mathcal{R}_{m_0,n_0}[\boldsymbol{A}\otimes\boldsymbol{B}]$ has the simple structure of a rank-1 matrix. Under a wrong configuration $\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]$ no

longer takes any special form. Nevertheless, the following lemma characterizes how the spectral norm of $\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]$ depends on further rearrangements of both $\boldsymbol{A}$ and $\boldsymbol{B}$. It is a property of the Kronecker products and the KPD (8.3), so we present it in the general form, without referring to any "true" configuration.

Lemma 9.1. *Let $\boldsymbol{A}$ be a $2^m\times 2^n$ matrix and $\boldsymbol{B}$ be a $2^{m^\dagger}\times 2^{n^\dagger}$ matrix. Then for any $m',n'\in\mathbb{Z},\ 0\leqslant m'\leqslant M,\ 0\leqslant n'\leqslant N$,*

$$\|\mathcal{R}_{m',n'}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S=\|\mathcal{R}_{m\wedge m',n\wedge n'}[\boldsymbol{A}]\|_S\cdot\|\mathcal{R}_{(m'-m)_+,(n'-n)_+}[\boldsymbol{B}]\|_S$$

Applying Lemma 9.1 to Example 9.1 leads to the following corollary.

Corollary 9.2. *For Example 9.1, under Assumption 9.1, it holds that*

$$\max_{(m,n)\in\mathcal{W}}\|\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S=\frac{1}{\sqrt{2}}+o_p(1).$$

And as a consequence, Assumption 9.4 holds with the λ_0 in (9.16) and $\psi_0^2=1/2$.

9.3.4 Multi-term Extension

For ease of presentation, we only provide the result and analysis of the two-term model

$$\boldsymbol{Y}=\lambda_1\boldsymbol{A}_1\otimes\boldsymbol{B}_1+\lambda_2\boldsymbol{A}_2\otimes\boldsymbol{B}_2+\frac{\sigma}{2^{(M+N)/2}}\boldsymbol{E}. \tag{9.17}$$

Similar results can be directly extended to the multi-term model.

We propose to use the same configuration selection procedure in Section 9.2.3, that is, for any candidate configuration $(m,n)\in\mathcal{C}$, although $\boldsymbol{Y}$ is generated from the two-term model (9.17), we nonetheless still calculate the information criterion (9.6) by fitting the one-term Kronecker product model (9.1) to $\boldsymbol{Y}$. This approach avoids the need of the determination of the number of Kronecker product terms when seeking the correct configuration. It allows the separation of the two. In this case, the estimated $\hat{\lambda}$ used in the information criterion (9.8) is

$$\hat{\lambda}=\|\mathcal{R}_{m,n}[\boldsymbol{Y}]\|_S=\|\lambda_1\mathcal{R}_{m,n}[\boldsymbol{A}_1\otimes\boldsymbol{B}_2]+\lambda_2\mathcal{R}_{m,n}[\boldsymbol{A}_2\otimes\boldsymbol{B}_2]+\sigma 2^{-(M+N)/2}\mathcal{R}_{m,n}[\boldsymbol{E}]\|_S. \tag{9.18}$$

Note that under the true configuration, we have $\hat{\lambda} \approx \lambda_1$. To bound $\hat{\lambda}$ under wrong configurations, we define

$$\phi_1 = \max_{(m,n)\in\mathcal{W}} \|\mathcal{R}_{m,n}[\boldsymbol{A}_1 \otimes \boldsymbol{B}_1]\|_S, \qquad \phi_2 = \max_{(m,n)\in\mathcal{W}} \|\mathcal{R}_{m,n}[\boldsymbol{A}_2 \otimes \boldsymbol{B}_2]\|_S,$$

and the representation gaps

$$\psi_1^2 := 1 - \phi_1^2, \qquad \psi_2^2 := 1 - \phi_2^2.$$

Even though $\mathrm{vec}(\boldsymbol{A}_1)$ and $\mathrm{vec}(\boldsymbol{A}_2)$ are orthogonal according to the model assumption, the column spaces of $\mathcal{R}_{m,n}[\boldsymbol{A}_1 \otimes \boldsymbol{B}_1]$ and $\mathcal{R}_{m,n}[\boldsymbol{A}_2 \otimes \boldsymbol{B}_2]$ are not necessarily orthogonal. In the worst case when $\mathcal{R}_{m,n}[\boldsymbol{A}_1 \otimes \boldsymbol{B}_1]$ and $\mathcal{R}_{m,n}[\boldsymbol{A}_2 \otimes \boldsymbol{B}_2]$ have the same column space and the same row space, then $\hat{\lambda}$ in (9.18) is close to $\lambda_1\phi_1 + \lambda_2\phi_2$, which may exceed λ_1. Therefore, we need to bound the distance between the column (and row) spaces of $\mathcal{R}_{m,n}[\boldsymbol{A}_1 \otimes \boldsymbol{B}_1]$ and $\mathcal{R}_{m,n}[\boldsymbol{A}_2 \otimes \boldsymbol{B}_2]$. For this purpose, we make use of the principal angles between linear subspaces. Specifically, if $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are two matrices of the same number of rows, the smallest principal angle between their column spaces, denote by $\Theta(\boldsymbol{M}_1, \boldsymbol{M}_2)$, is defined as

$$\cos\Theta(\boldsymbol{M}_1, \boldsymbol{M}_2) = \sup_{u_1\neq 0, u_2\neq 0} \frac{u_1'\boldsymbol{M}_1'\boldsymbol{M}_2 u_2}{\|\boldsymbol{M}_1 u_1\|\|\boldsymbol{M}_2 u_2\|}.$$

We first discuss the deterministic scheme, where $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are non-random. In Assumption 9.5, θ_c and θ_r are lower bounds of the smallest possible principal angles between the column spaces and the row spaces of the two rearranged components, respectively.

Assumption 9.5. *There exist* $0 < \xi < 1$ *such that*

$$\max_{(m,n)\in\mathcal{W}_A} \cos\Theta(\mathcal{R}_{m,n}[\boldsymbol{A}_1 \otimes \boldsymbol{B}_1], \mathcal{R}_{m,n}[\boldsymbol{A}_2 \otimes \boldsymbol{B}_2]) \leqslant \xi,$$

and

$$\max_{(m,n)\in\mathcal{W}_B} \cos\Theta([\mathcal{R}_{m,n}[\boldsymbol{A}_1 \otimes \boldsymbol{B}_1]]', [\mathcal{R}_{m,n}[\boldsymbol{A}_2 \otimes \boldsymbol{B}_2]]') \leqslant \xi,$$

where

$$\mathcal{W}_A = \{(m,n) \in \mathcal{W} : m+n \geqslant m^\dagger + n^\dagger\}, \mathcal{W}_B = \{(m,n) \in \mathcal{W} : m+n < m^\dagger + n^\dagger\}.$$

The following lemma provides an upper bound of the spectral norm of a sum of two matrices. It utilizes the principal angles between the column and row spaces to make the bound sharper than the one given by the triangular inequality. Assumption 9.5 enables us to apply Lemma 9.2 to bound $\hat{\lambda}$ in (9.18).

Lemma 9.2. *Suppose $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are two matrices of the same dimension. Let $\|\boldsymbol{M}_1\|_S = \mu$, $\|\boldsymbol{M}_2\|_S = \nu$. Denote the principle angles between the column spaces and the row spaces as $\theta = \Theta(\boldsymbol{M}_1, \boldsymbol{M}_2)$, $\eta = \Theta(\boldsymbol{M}_1', \boldsymbol{M}_2')$, respectively. Then*

$$\|\boldsymbol{M}_1 + \boldsymbol{M}_2\|_S^2 \leqslant \Lambda^2(\mu, \nu, \theta, \eta),$$

where

$$\Lambda^2(\mu,\nu,\theta,\eta) = \frac{1}{2}\Big[\sqrt{(\mu^2+\nu^2+2\mu\nu\cos\theta\cos\eta)^2 - 4\mu^2\nu^2\sin^2\theta\sin^2\eta} + \mu^2+\nu^2+2\mu\nu\cos\theta\cos\eta\Big].$$

Similar to Assumption 9.3, we assume the signal strengths λ_1, λ_2 and the noise level σ satisfy the following assumption.

Assumption 9.6. *For model (9.17), we assume that λ_k and the matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$, $k = 1, 2$ are deterministic and*

$$\lim_{M+N\to\infty} \frac{2^{M+N}}{2^{m+n}+2^{m^\dagger+n^\dagger}} \frac{\lambda_1^2\psi_1^2 - \lambda_2^2\phi_2^2 - 2\lambda_1\lambda_2\phi_1\phi_2\xi}{\sigma^2+\lambda_2^2} = \infty \tag{9.19}$$

and

$$\lim_{M+N\to\infty} 2^{(M+N)/4} \frac{\lambda_1^2\psi_1^2 - \lambda_2^2\phi_2^2 - 2\lambda_1\lambda_2\phi_1\phi_2\xi}{(\lambda_1+\lambda_2)\sigma} = \infty. \tag{9.20}$$

The conditions (9.19) and (9.20) correspond to (9.11) and (9.12) in the one-term model. Specifically, when $\lambda_2 = 0$, the two-term model reduces to one-term case and Assumption 9.6

reduces to Assumption 9.3 as well. The main result for the two-term model is stated in Theorem 9.5.

Theorem 9.5. *Consider the two-term model* (9.17)*, where* λ_k *and the matrices* $\boldsymbol{A}_k$ *and* $\boldsymbol{B}_k$ *are deterministic. Suppose Assumptions 9.1, 9.2, 9.5 and 9.6 hold. If* κ *satisfies*

$$\kappa \geqslant 2\ln 2 \quad \textit{and} \quad \kappa = o\left(\frac{2^{M+N}\alpha}{2^{m_0+n_0}+2^{M+N-m_0-n_0}}\right),$$

then

$$\min_{(m,n)\in\mathcal{W}} \mathbb{E}[\mathrm{IC}_\kappa(m,n)] - \mathbb{E}[\mathrm{IC}_\kappa(m_0,n_0)] \geqslant 2^{M+N}\alpha(1+o_p(1)),$$

where

$$\alpha = \ln\left(1+\frac{\lambda_1^2\psi_1^2-\lambda_2^2\phi_2^2-2\lambda_1\lambda_2\phi_1\phi_2\xi}{\sigma^2+\lambda_2^2}\right). \tag{9.21}$$

Furthermore, the consistency (9.14) *continues to hold.*

Similar to Theorem 9.2, we have shown that for the two-term model, the information criterion obtained by fitting a one-term model can still separate the true and wrong configurations with a gap of the order $O(2^{M+N}\alpha)$. On the other hand, comparing with Assumption 9.3, Theorem 9.5 depends on Assumption 9.6, which requires not only the signal-to-noise ratio (λ_1/σ), but also the relative strength of the two terms (λ_1/λ_2) to be large enough. Comparing the two term model (9.17) with the one term model (i.e. $\lambda_2 = 0$), we note that the information criterion gap α in Theorem 9.5 is smaller than the one given by Theorem 9.2. This phenomenon can be intuitively explained through (9.21). On one hand, λ_2^2 contributes to the noise term in identifying the configuration of the first Kronecker product as $\lambda_2^2+\sigma^2$ appears in the denominator in (9.21). On the other hand, over-fitting due to the second Kronecker product reduces $\|\boldsymbol{Y}-\hat{\boldsymbol{Y}}\|_F^2$ under the wrong configuration, which is quantified by $\lambda_2^2\phi_2^2+2\lambda_1\lambda_2\phi_1\phi_2\xi$ in the numerator of (9.21).

Similar to Example 9.1, we consider the following example of the two term model under normality.

Example 9.2. *Consider the two term model* (9.17)*. Suppose that*

$$\lambda_k\boldsymbol{A}_k\otimes\boldsymbol{B}_k = \lambda_{k0}\tilde{\boldsymbol{A}}_k\otimes\tilde{\boldsymbol{B}}_k/2^{(M+N)/2}, \quad k=1,2,$$

where all of the five matrices $\tilde{\boldsymbol{A}}_k$ *and* $\tilde{\boldsymbol{B}}_k$ *and* $\boldsymbol{E}$ *are independent, and each consisting of IID standard normal entries. To translate it back into the form of* (9.17)*, we let* $\boldsymbol{A}_k = \tilde{\boldsymbol{A}}_k/\|\tilde{\boldsymbol{A}}_k\|_F$, $\boldsymbol{B}_k = \tilde{\boldsymbol{B}}_k/\|\tilde{\boldsymbol{B}}_k\|_F$, *and* $\lambda_k = \lambda_{k0} \cdot \|\tilde{\boldsymbol{A}}_k\|_F \cdot \|\tilde{\boldsymbol{B}}_k\|_F/2^{(M+N)/2}$.

For Example 9.2, it turns out that with probabilities tending to one, ξ is close to 0 and the representation gaps ψ_1^2 and ψ_2^2 are close to 1/2 (due to Corollary 9.2). As an immediate consequence, Theorem 9.5 yields a information criterion gap of the size

$$\alpha = \ln\left(1 + \frac{\lambda_{10}^2 - \lambda_{20}^2}{2(\sigma^2 + \lambda_{20}^2)}\right).$$

However, by a refined analysis of Assumption 5 under the normality of Example 9.2, we are able to prove the following improved result.

Corollary 9.3. *Consider Example 9.2. Under Assumptions 9.1 and 9.2, Theorem 9.5 holds with the information criterion gap*

$$\alpha = \ln\left(1 + \frac{\lambda_{10}^2}{2(\sigma^2 + \lambda_{20}^2)}\right).$$

9.4 Simulations and Examples

9.4.1 Simulations

We design two simulation studies: the first one on the performance of the estimation procedure introduced in Section 9.2.2, and the second one on the configuration selection proposed in Section 9.2.3. Many implications of the theoretical results in Section 9.3 are demonstrated in the outcome of the numerical studies.

Estimation with known configuration

We first consider the performance of the estimators of λ, $\boldsymbol{A}$ and $\boldsymbol{B}$ given in (9.4), when the true configuration (m_0, n_0) is known. Throughout this subsection the simulations are based on model (9.1) with $m_0 = 5, n_0 = 5, M = 10, N = 10$ and $\sigma = 1$.

The model (9.1) after the rearrangement under the true configuration becomes

$$\mathcal{R}_{m_0,n_0}[\boldsymbol{Y}] = \lambda \mathrm{vec}(\boldsymbol{A})\mathrm{vec}(\boldsymbol{B})' + \sigma 2^{-(M+N)/2}\mathcal{R}_{m_0,n_0}[\boldsymbol{E}],$$

where $\mathrm{vec}(\boldsymbol{A}) \in \mathbb{R}^{2^{m_0+n_0}}$, $\mathrm{vec}(\boldsymbol{B}) \in \mathbb{R}^{2^{m_0^\dagger+n_0^\dagger}}$ are unit vectors. Without loss of generality, set $\mathrm{vec}(\boldsymbol{A}) = (1,0,\ldots,0)'$, $\mathrm{vec}(\boldsymbol{B}) = (1,0,\ldots,0)'$. In this experiment, the noise level is fixed at $\sigma = 1$, so the signal-to-noise ratio is controlled by λ, which takes values from the set $\{e^1, e^2, \ldots, e^{16}\}$. For each value of λ, we calculate the errors of the corresponding estimators $\hat{\lambda}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ by

$$\ln\left(\frac{\hat{\lambda}}{\lambda} - 1\right)^2 \quad \text{and} \quad \ln\|\hat{\boldsymbol{A}} - \boldsymbol{A}\|_F^2 + \ln\|\hat{\boldsymbol{B}} - \boldsymbol{B}\|_F^2.$$

The errors based on 20 repetitions are reported in Figure 9.1.

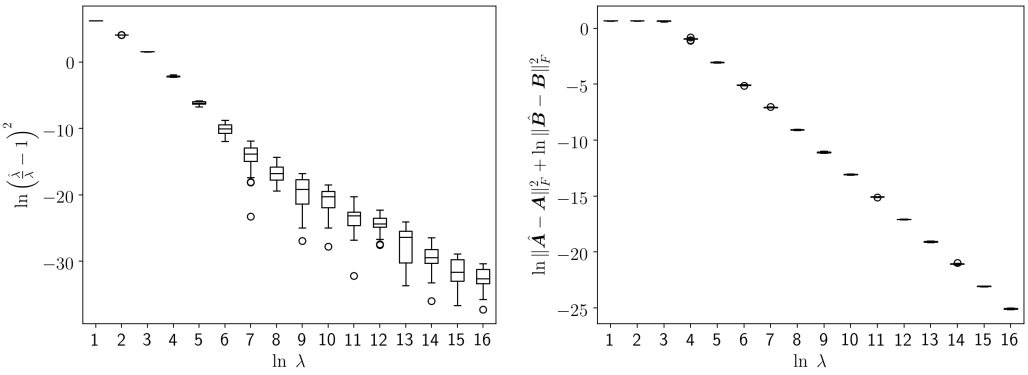


Figure 9.1: Boxplots for errors in $\hat{\lambda}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ against the signal-to-noise ratio.

Figure 9.1 displays an interesting linear pattern, that is, as the signal-to-noise ratio increases, $\ln\left(\frac{\hat{\lambda}}{\lambda} - 1\right)^2$ is approximately linear against $\ln\lambda$ with a slope around -2, and so is the error $\ln(\|\hat{\boldsymbol{A}} - \boldsymbol{A}\|_F^2\|\hat{\boldsymbol{B}} - \boldsymbol{B}\|_F^2)$ for the matrix estimators. We note that this pattern is consistent with Theorem 9.1, which asserts that

$$\frac{\hat{\lambda}}{\lambda} - 1 = O_p\left(\frac{1}{\lambda}\right) \quad \text{and} \quad \|\hat{\boldsymbol{A}} - \boldsymbol{A}\|_F\|\hat{\boldsymbol{B}} - \boldsymbol{B}\|_F = O_p\left(\frac{1}{\lambda}\right),$$

since r_0 defined in Theorem 9.1 remains a constant here as we vary the signal strength λ in the simulation.

Configuration Selection

We now demonstrate the performance of the information criterion based procedure for selecting the configuration. Two criteria will be considered: MSE (when $\kappa = 0$) and AIC (when $\kappa = 2$). Corresponding to the one- and multi-term models considered in Sections 9.3 and 9.2.4, we carry out two experiments respectively.

Experiment 1: One-term KoPA model

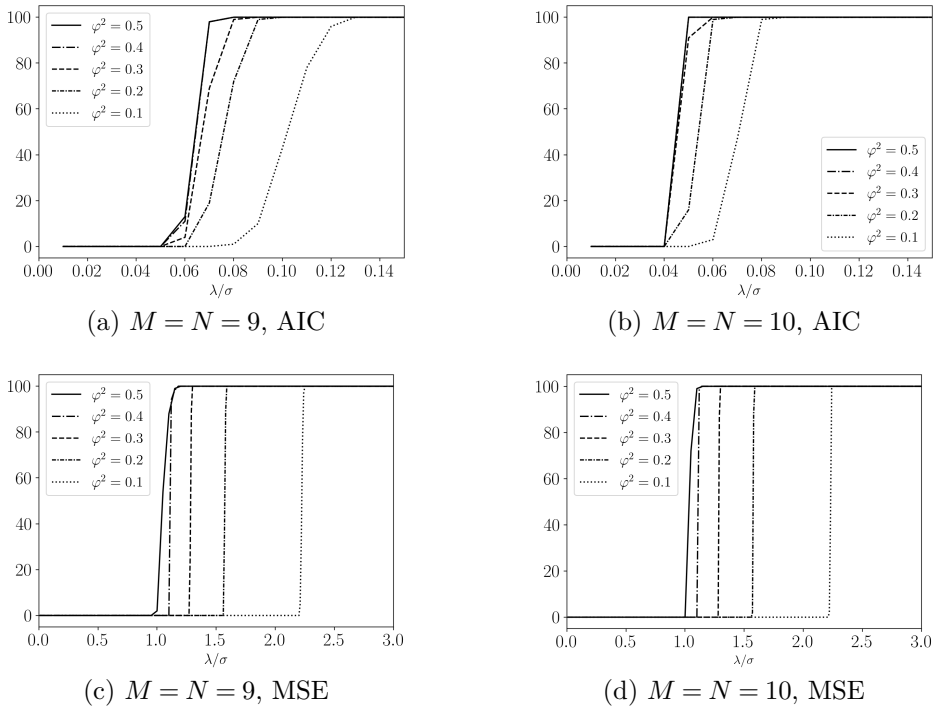
The simulation is based on model (9.1). Two configurations are considered: (i) $M = N = 9$, $m_0 = 4$, $n_0 = 4$, and (ii) $M = N = 10$, $m_0 = 5$, $n_0 = 4$. Similar to Section 9.4.1, the noise level is fixed at $\sigma = 1$, so the signal-to-noise ratio is controlled by λ. To control the representation gap ψ^2, we construct the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as follows:

$$\boldsymbol{A} = \sqrt{\varphi^2}\begin{bmatrix}1\\0\end{bmatrix} \otimes \boldsymbol{D}_1 + \sqrt{1-\varphi^2}\begin{bmatrix}0\\1\end{bmatrix} \otimes \boldsymbol{D}_2,$$
$$\boldsymbol{B} = \sqrt{\varphi^2}\begin{bmatrix}1\\0\end{bmatrix} \otimes \boldsymbol{D}_3 + \sqrt{1-\varphi^2}\begin{bmatrix}0\\1\end{bmatrix} \otimes \boldsymbol{D}_4,$$

where $\mathrm{vec}(\boldsymbol{D}_i), i = 1,2,3,4$ are independent random unit vectors such that $\mathrm{vec}(\boldsymbol{D}_1)$ and $\mathrm{vec}(\boldsymbol{D}_2)$ are orthogonal, and so are $\mathrm{vec}(\boldsymbol{D}_3)$ and $\mathrm{vec}(\boldsymbol{D}_4)$. In the experiment, five values of φ^2 are considered: $\varphi^2 \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. We remark that the construction above controls the representation gaps for configurations $(1,0)$ and (m_0+1, n_0) at φ^2 exactly, and the representation gaps for configurations with $m+n \in \{1, M+N-1\}$ (close to trivial configurations) or $|m-m_0|+|n-n_0| = 1$ (close to the true configuration) at roughly 0.5. Consequently, when $\varphi^2 = 0.1, 0.2, 0.3, 0.4$, the overall representation gap ψ^2 is at the desired level φ^2 with high probabilities. But when $\varphi^2 = 0.5$, the representation gap ψ^2 can be slightly smaller than 0.5.

In Figure 9.2, we plot the empirical frequencies of the correct configuration selection, out of 100 repetitions, against the signal-to-noise ratio λ/σ. Note that the x-axis scale in subfigures 9.2a and 9.2b is different from that in 9.2c and 9.2d. The performances of both MSE ($\kappa = 0$) and AIC ($\kappa = 2$) are illustrated. BIC ($\kappa = (M+N)\ln 2$) has a very similar

performance to AIC, and is not reported here.



(a) $M = N = 9$, AIC

(b) $M = N = 10$, AIC

(c) $M = N = 9$, MSE

(d) $M = N = 10$, MSE

Figure 9.2: The empirical frequencies of the correct configuration selection out of 100 repetitions.

For extremely weak signal-to-noise ratio $\lambda \leqslant 0.03$, neither of MSE and AIC is able to select the true configuration with a high probability, for both configurations. This does not contradict with Theorem 9.3. When the signal is very weak, larger dimensions of the observed matrix $\boldsymbol{Y}$ are required for the consistency. As the signal-to-noise ratio increases from 0.01 to 0.13, the probability that the true configuration is selected increases gradually and eventually gets very close to one for AIC as shown in Figures 9.2a and 9.2b. We also note that the performance gets better as the representation gap ψ^2 increases. These observations are echoing Theorem 9.2, which shows that AIC (with $\kappa = 2 > 2\ln 2$) only requires a minimal condition $(\lambda/\sigma)^2\psi^2 > 0$ to achieve the consistency, and the separation gap of AIC is a monotone function of $(\lambda/\sigma)^2\psi^2$. On the other hand, the performance of MSE exhibits a phase transition: it only starts to select the true configuration with a decent probability when the signal-to-noise ratio λ/σ exceed a certain threshold. The theoretical asymptotic threshold for MSE is $\lambda/\sigma \geqslant \sqrt{1/(2\psi^2)}$ as discussed in Remark 5. For $\psi^2 \in \{0.5, 0.4, 0.3, 0.2, 0.1\}$ used in this simulation, the corresponding thresholds for λ/σ are

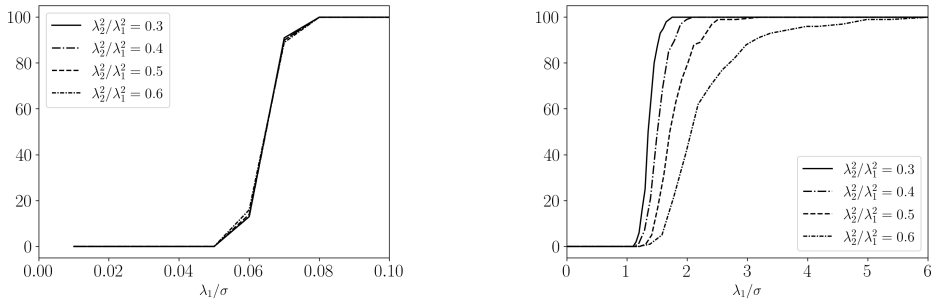
$\{1, 1.12, 1.29, 1.58, 2.24\}$, which can be clearly visualized in Figures 9.2c and 9.2d.

Comparing Figures 9.2a with Figures 9.2b, we see that the empirical frequency curve increases from 0 to 100 much faster when the matrices are larger. This is consistent with Theorem 9.2, which shows that the probability of correct configuration selection approaches 1 exponentially fast.

Experiment 2: Two-term KoPA model

In the second experiment, we consider the two-term KoPA model in (9.17) where $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are generated under the random scheme in Example 9.2 such that $\psi_1^2 \approx 1/2$, $\psi_2^2 \approx 1/2$ and $\xi \approx 0$. According to Theorem 9.5, besides the signal-to-noise ratio λ_1/σ, the relative strength of the second term λ_2/λ_1 (for the random scheme adopted in this experiment, see Corollary 9.3) affects the configuration selection as well.

In this simulation, we fix the configurations to $M = N = 9, (m_0, n_0) = (4, 4)$ and consider four different relative strengths of the second term $\lambda_2^2/\lambda_1^2 \in \{0.3, 0.4, 0.5, 0.6\}$. Similar to Experiment 1, we report the empirical frequencies of correct configurations selection of MSE and AIC, out of 100 repetitions, as a function of the signal-to-noise ratio λ_1/σ in Figure 9.3.



(a) $M = N = 9$, AIC (b) $M = N = 9$, MSE

Figure 9.3: The empirical frequencies of the correct configuration selection out of 100 repetitions in a two-term model.

Figure 9.3a shows that the performance of AIC is in-sensitive to the ratio λ_2^2/λ_1^2 over the experimented range. To the contrary, it is seen from Figure 9.3b that MSE performs

better when the ratio λ_2^2/λ_1^2 gets smaller, which is consistent with Corollary 9.3.

9.4.2 Analysis on Images

The cameraman's image

In this section we analyze the famous cameraman image. The original image, denoted by $\boldsymbol{Y}_0$, has 512×512 pixels. Each entry of $\boldsymbol{Y}_0$ is a real number between 0 and 1, where 0 codes black and 1 indicates white. The grayscale cameraman image $\boldsymbol{Y}_0$ is displayed in Figure 9.4.

Figure 9.4: The cameraman image.

Our analysis will be based on the de-meaned version $\boldsymbol{Y}$ of the original image $\boldsymbol{Y}_0$. We demonstrate how well the image $\boldsymbol{Y}$ can be approximated by a Kronecker product or the sum of a few Kronecker products, and make comparisons with the low rank approximations given by SVD.

We first consider the configuration selection by MSE, AIC and BIC on the original image $\boldsymbol{Y}$. Figure 9.5 plots the heat maps for the information criterion $\mathrm{IC}_\kappa(m,n)$ for all candidate configurations in the set

$$\mathcal{C} = \{(m,n) : \, 0 \leqslant m,n \leqslant 9\} \setminus \{(0,0),(9,9)\},$$

where the top-left and bottom-right corners are always excluded from the consideration. Since darker cells correspond to smaller values of the information criteria, we see that MSE and AIC select the configuration $(8,9)$, and BIC selects $(6,7)$.

We also observe an overall pattern in Figure 9.5: configurations with larger (m,n) values are more preferable than those with smaller (m,n). Note that the Kronecker product does

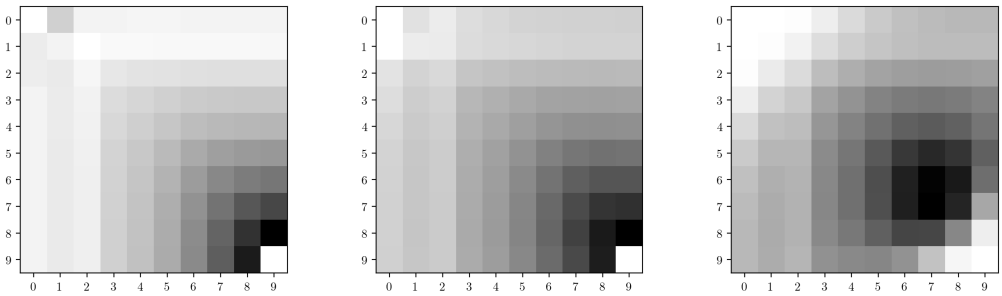


Figure 9.5: Information Criteria for the cameraman's image. (Left) MSE (Mid) AIC (Right) BIC. Darker color corresponds to lower IC value.

not commute, and with configuration (m,n) the product is a $2^m \times 2^n$ block matrix, each block of the size $2^{9-m} \times 2^{9-n}$. Real images usually show the locality of pixels in the sense that nearby pixels tend to have similar colors. Therefore, it can be understood that larger values of m and n are preferred, since they are better suited to capture the locality. Actually, for the cameraman's image, the configuration $(8,9)$ accounts for 99.50% of the total variation of $\boldsymbol{Y}$. The penalty on the number of parameters in AIC is not strong enough to offset the closer approximation given by the configuration $(8,9)$. With a stronger penalty term, BIC selects a configuration that is closer to the center of the configuration space, involving a much smaller number of parameters.

From the perspective of image compressing, KoPA is more flexible than the low rank approximation, by allowing a choice of the configuration, and hence a choice of the compression rate. To compare their performances, we use the ratio $\|\hat{\boldsymbol{Y}}\|_F^2/\|\boldsymbol{Y}\|_F^2$ to measure how close the approximation $\hat{\boldsymbol{Y}}$ is to the original image $\boldsymbol{Y}$. In Figure 9.6, these ratios are plotted against the numbers of parameters for the KPD, marked by "+" on the solid line. Since the number of parameters involved in the Kronecker product with configuration (m,n) is $\eta = 2^{m+n} + 2^{M+N-m-n}$, the configurations $\{m,n : m+n = c\}$ for any given $0 < c < M+N$ have the same number of parameters. Among these configurations, we only plot the one with the largest $\|\hat{\boldsymbol{Y}}\|_F^2/\|\boldsymbol{Y}\|_F^2$. On the other hand, each cross stands for a rank-k approximation of $\boldsymbol{Y}$, where its value on the horizontal axis is the number of parameters

$$\eta = 1 + \sum_{j=1}^{k} (2^M + 2^N - 2j + 1) \quad \text{for } k = 1, \ldots, 2^{M \wedge N}.$$

According to Figure 9.6, there always exists a one-term Kronecker product which provides a better approximation of the original cameraman's image than the best low rank approximation involving the same number of parameters.

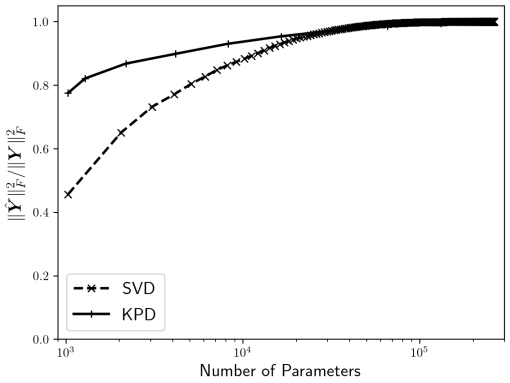


Figure 9.6: Percentage of variance explained against number of parameters, for KoPA with all configurations, and for low rank approximations of all ranks.

We also consider de-noising the images corrupted by additive Gaussian white noise

$$\boldsymbol{Y}_\sigma = \boldsymbol{Y} + \sigma \boldsymbol{E},$$

where $\boldsymbol{E}$ is a matrix with IID standard normal entries. We experiment with three levels of corruption: $\sigma = 0.1, 0.2, 0.3$. Examples of the corrupted images with different σ are shown in Figure 9.7 with the values rescaled to $[0,1]$ for plotting.

For the corrupted images, the information criteria $\mathrm{IC}_\kappa(m,n)$ are calculated, and the corresponding heat maps are plotted in Figure 9.8. With added noise, AIC and BIC tend

Figure 9.7: Noisy cameraman's images when (Left) $\sigma = 0.1$ (Mid) $\sigma = 0.2$ (Right) $\sigma = 0.3$

to select configurations in the middle of the configuration space.

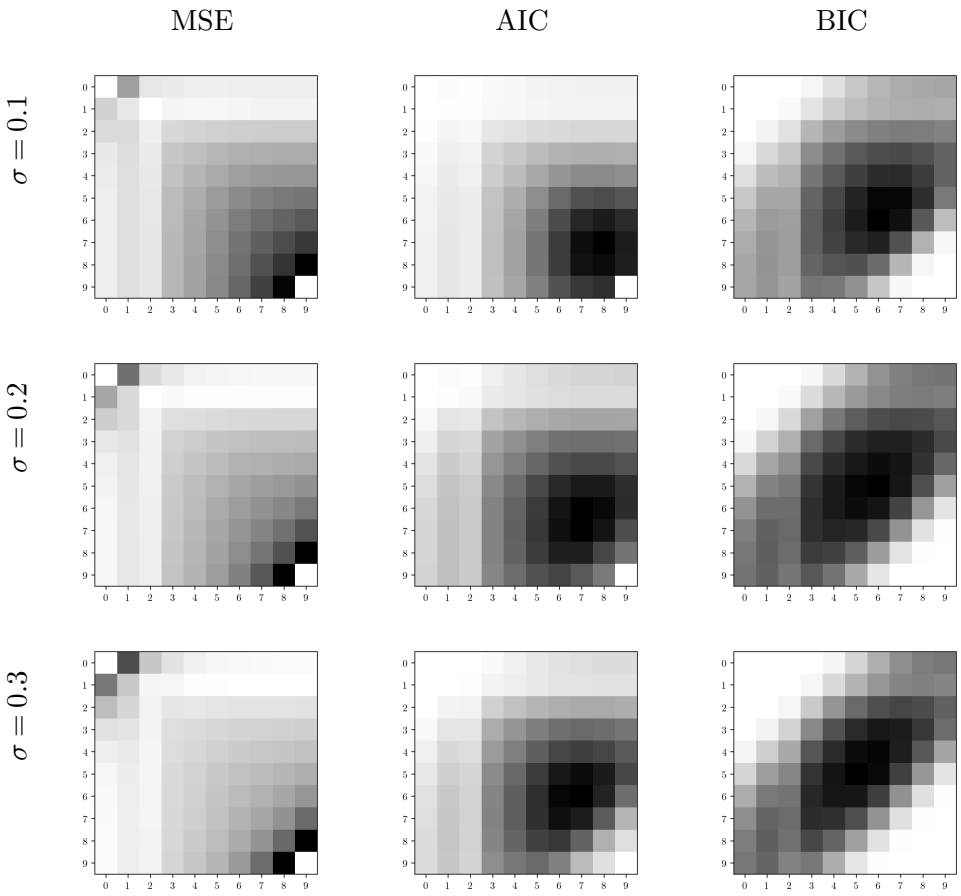


Figure 9.8: Heat maps for three different information criteria for the camera's images with different noise levels. Darker color means lower IC value.

Now we consider multi-term Kronecker approximation. Following the discussion in Section 9.2.4, for each of three corrupted images $\boldsymbol{Y}_\sigma$, we use the configuration selected by BIC in Figure 9.8. Specifically, configurations $(6,6)$, $(5,6)$ and $(5,5)$ are selected when $\sigma = 0.1$, 0.2 and 0.3, respectively. A two-term Kronecker product model (9.17) is then fitted under the selected configuration. The fitted images are plotted in the upper panel of Figure 9.9. Each of them is compared with the image obtained by the low rank approximation involving a similar number of parameters as the two-term KoPA. From Figure 9.9, it is quite evident that the image details can easily be recognized from the images reconstructed from two-term KoPA, but can hardly be perceived in those given by the low rank approximation.

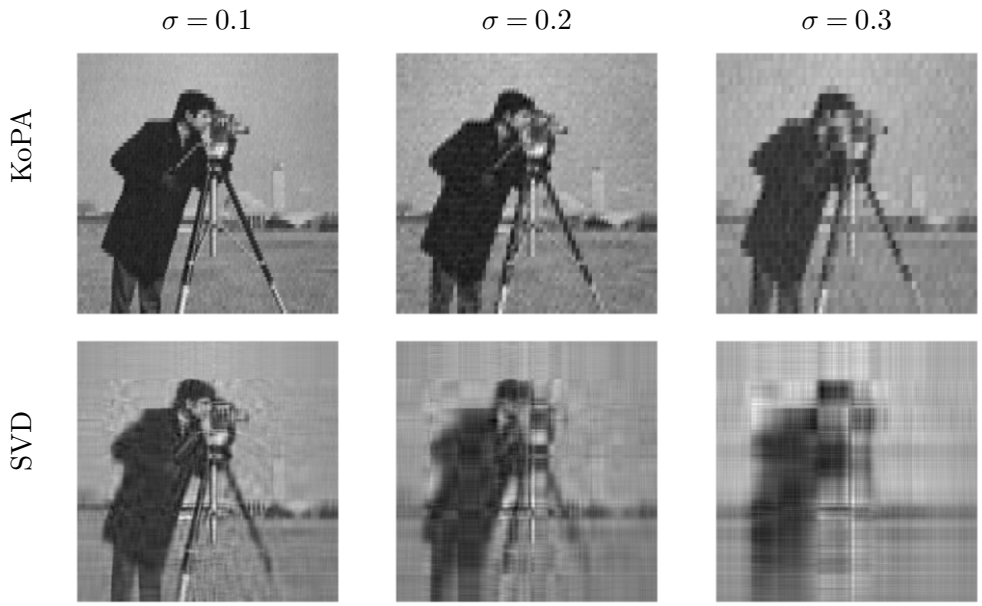


Figure 9.9: The fitted image given by multi-term KoPA, and the SVD approximation with similar number of parameters.

Finally, we examine the reconstruction error defined by

$$\frac{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|_F^2}{\|\boldsymbol{Y}\|_F^2},$$

where $\boldsymbol{Y}$ is the original image and $\hat{\boldsymbol{Y}}$ is the one reconstructed from $\boldsymbol{Y}_\sigma$. For each of the three noisy images, we continue to use the configuration selected by BIC. With fixed configurations, we keep increasing the number of terms in the KoPA until $\boldsymbol{Y}_\sigma$ is fully fitted, and plot the corresponding reconstruction error against the number of parameters in Figure 9.10. It has the familiar "U" shape, showing the trade-off between estimation bias and variation. A similar curve is given for the low rank approximations exhausting all possible ranks. From Figure 9.10, it is seen that the multi-term KoPA constantly outperforms the low rank approximation at any given number of parameters. Furthermore, the minimum reconstruction error that KoPA can reach is always smaller than that given by the low rank approximation.

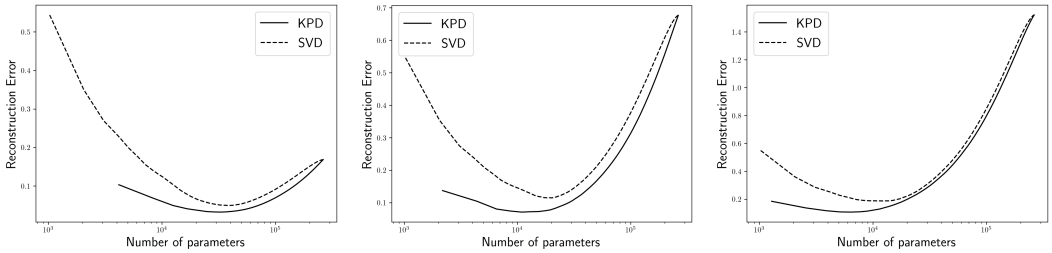


Figure 9.10: Reconstruction error against the number of parameters for KoPA and low rank approximations. The three panels from left to right correspond to $\sigma = 0.1$, $\sigma = 0.2$ and $\sigma = 0.3$ respectively.

Figure 9.11: List of test images.

More images

To assess the performance of KoPA model in image denoising, we repeat the experiment in Section 9.4.2 to a larger set of test images. The 10 test images printed in Figure 9.11 are collected from Image Processing Place[1] and The Waterloo image Repository[2]. Each of the 10 test images is a 512×512 gray-scaled matrix, same as the cameraman's image. We corrupt the test image with additive Gaussian noise, whose amplitude is given by 0.5 times the standard deviation of all its pixel values:

$$\boldsymbol{Y}_\sigma = \boldsymbol{Y} + 0.5 \cdot \mathrm{std}(\boldsymbol{Y}) \cdot \boldsymbol{E}.$$

We compare five methods of denoising these images: one-term SVD and KoPA mod-

[1] http://www.imageprocessingplace.com/root_files_V3/image_databases.htm
[2] http://links.uwaterloo.ca/Repository.html

image	SVD	KoPA	mSVD	mKoPA	TVR
boat	0.4709	0.1757	0.0853	0.0613	0.0356
cameraman	0.5446	0.1337	0.0644	0.0399	0.0294
goldhill	0.4632	0.1391	0.0759	0.0568	0.0363
jetplane	0.7347	0.1853	0.0866	0.0596	0.0302
lake	0.5425	0.1287	0.0825	0.0539	0.0308
livingroom	0.6747	0.2055	0.0995	0.0811	0.0589
mandril	0.6949	0.3557	0.1471	0.0889	0.0739
peppers	0.7394	0.1075	0.0734	0.0445	0.0224
pirate	0.7746	0.1533	0.1018	0.0686	0.0413
walkbridge	0.6617	0.2085	0.1263	0.0925	0.0593

Table 9.1: Reconstruction errors of one-term SVD, one-term KoPA, multi-term SVD(mSVD), multi-term KoPA(mKoPA) and total variation regularization (TVR) on the ten test images.

els, multi-term SVD and KoPA models, image denoising algorithm through total variation regularization (Chambolle, 2004). Since determining the number of terms in multi-term models is beyond the scope of this article, the number of terms in the multi-term models are chosen to minimize the reconstruction error. The performance of the five approaches on the ten images are reported in Table 9.1.

For each image, the configuration of the KoPA is selected by BIC ($\kappa = 18 \ln 2$). From Table 9.1, the KoPA-based methods outperform SVD-based approaches, which is not surprising as SVD corresponds to a special configuration in KoPA models. On the other hand, the image denoising based on KoPA (and multi-term KoPA) is close to the TVR (total variation regularization) method but the latter does have a superior performance.

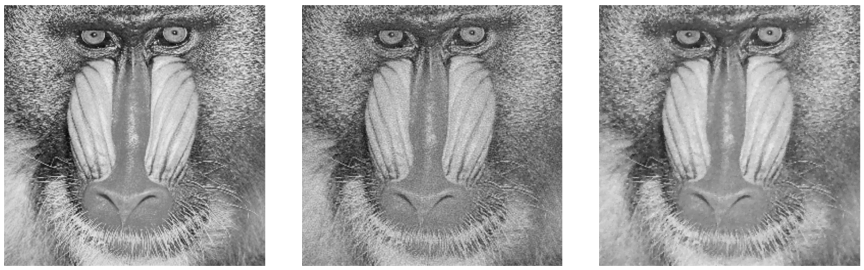


Figure 9.12: (left) The mandrill image, (mid) recovered images from multi-term KoPA model and (right) total variation regularization.

We note that KoPA and TVR are not directly comparable. Image is a special type of matrix data, whose entries usually possess certain continuity in values. TVR fully utilizes

this continuity by imposing regularization on the total variation while SVD and KoPA do not. The difference can be seen from Figure 9.12 as well. The TVR can recover the smooth region (the mandrill's nose) well, while the multi-term KoPA model has more details in non-smooth regions (the mandrill's fur and beard). Finally we remark that the performance of KoPA approach on image analysis can possibly be improved by adding a similar penalty term on the smoothness of $\boldsymbol{B}$.

CHAPTER 10

Hybrid Kronecker Product Approximation

10.1 Introduction

It is often the case that KoPA using a single configuration requires a large number of terms to make the approximation accurate. By allowing the use of a sum of Kronecker products of different configurations, an observed high dimensional matrix (image) can be approximated more effectively using a much smaller number of parameters (elements). We note that often the observed matrix can have much more complex structure than a single Kronecker product can handle. For example, representing an image with Kronecker products of the same configuration is often not satisfactory since the configuration dimensions determine the block structure of the recovered image, similar to the pixel size of the image. A single configuration is often not possible to provide as much detail as needed. Similar to the extension from low rank matrix approximation to KoPA of a single configuration, we propose to extend the Kronecker product approach to allow for multiple configurations. It is more flexible and may provide a more accurate representation with a smaller number of parameters.

In this chapter, we generalize the KoPA model in Chapter 9 to a multi-term setting, where the observed high dimensional matrix is assumed to be generated from a sum of several Kronecker products of different configurations – we name the model *hybrid* KoPA (*h*KoPA). As a special case, when all the Kronecker products are vector outer products, KoPA corresponds to a low rank matrix approximation.

We consider two problems in this chapter. We first propose a procedure to estimate a

hKoPA with a set of known configurations. The procedure is based on an iterative backfitting algorithm with the basic operation being finding the least squares solution of a single Kronecker product of a given configuration to a given matrix. This operation can be obtained through a rearrangement operation and a SVD estimation. Next, we consider the problem of determining the configurations in a hKoPA for a given observed matrix. As exploiting the space of all possible configuration combinations is computationally expensive, we propose an iterative greedy algorithm similar to the boosting algorithm (Freund et al., 1999). In each iteration, a single Kronecker product term is added to the model by fitting the residual matrix from the previous iteration. The configuration of the added Kronecker product is determined using the procedure proposed in Section 9.2.3. This algorithm efficiently fits a hKoPA model with a potentially sub-optimal solution as a compromise between computation and accuracy.

10.2 The Hybrid KoPA Model

We consider the K-term hKoPA model, where the observed matrix $\boldsymbol{Y}$ is of the form

$$\boldsymbol{Y} = \sum_{k=1}^{K} \lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k + \sigma \boldsymbol{E}. \tag{10.1}$$

In other words, $\boldsymbol{Y}$ is generated as the sum of K Kronecker products and a noise matrix. We assume $\boldsymbol{Y}$ is of the dimension $2^M \times 2^N$, and the matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ in the k-th component are $2^{m_k} \times 2^{n_k}$ and $2^{M-m_k} \times 2^{N-n_k}$, respectively. The matrix $\boldsymbol{E}$ is a noise matrix with i.i.d. standard Gaussian entries. The model and the methodology that we propose here also extend to $\boldsymbol{Y}$ of any dimensions, as long as the corresponding $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ have conformal dimensions. However, for simplicity, we will assume in this paper that all matrix dimensions are of powers of 2.

We define the configuration of the hKoPA model in (10.1) as the collection of individual configurations $\mathcal{C} := \{(m_k, n_k), 1 \leq k \leq K\}$. The Kronecker product components in (10.1) are allowed to have different configurations (m_k, n_k). When the model configuration $\mathcal{C}$ is known, we need to estimate the component matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$, for $k = 1, \ldots, K$ in model

(10.1). When the configuration $\mathcal{C}$ is unknown, the estimation of model (10.1) requires the determination of the configuration $\mathcal{C}$ as well, resulting in a configuration determination problem in addition to the estimation problem.

Some existing researches on Kronecker product structured data can be viewed as the special cases of model (10.1). When $K = 1$ and the configuration is unknown, model (10.1) reduces to the single-term KoPA model investigated in Section 9.2.1. When the configurations of the K Kronecker products are known and equal such that $(m_1, n_1) = (m_2, n_2) = \cdots = (m_K, n_K)$, the estimation of model (10.1) can also be obtained based on the Kronecker product decomposition introduced in Section 8.2.

The primary goal is to estimate λ_k, $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ in (10.1). For this purpose, we need some identifiability conditions. The first one takes care of the scaling of $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ so that they are determined up to a sign change.

Assumption 10.1 (Identifiability Condition 1)**.** *Assume*

$$\|\boldsymbol{A}_k\|_F = \|\boldsymbol{B}_k\|_F = 1, \text{ for } k = 1, \ldots, K,$$

and $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_K > 0$.

If $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ have the same dimensions (hence so do $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$), there is an obvious identifiability issue, since

$$\lambda_1 \boldsymbol{A}_1 \otimes (\boldsymbol{B}_1 + c\lambda_2 \boldsymbol{B}_2) + \lambda_2 (\boldsymbol{A}_2 - c\lambda_1 \boldsymbol{A}_1) \otimes \boldsymbol{B}_2$$

always gives the same sum for any $c \in \mathbb{R}$. There is another type of unidentifiability that is more subtle. Suppose $\boldsymbol{A}_1$ is *smaller* $\boldsymbol{A}_2$ in the sense that $m_1 \leqslant m_2$ and $n_1 \leqslant n_2$, then for any $2^{m_2 - m_1} \times 2^{n_2 - n_1}$ matrix $\boldsymbol{C}$, it holds that

$$\lambda_1 \boldsymbol{A}_1 \otimes (\boldsymbol{B}_1 + \lambda_2 \boldsymbol{C} \otimes \boldsymbol{B}_2) + \lambda_2 (\boldsymbol{A}_2 - \lambda_1 \boldsymbol{A}_1 \otimes \boldsymbol{C}) \otimes \boldsymbol{B}_2 = \lambda_1 \boldsymbol{A}_1 \otimes \boldsymbol{B}_1 + \lambda_2 \boldsymbol{A}_2 \otimes \boldsymbol{B}_2, \quad (10.2)$$

Therefore, we also make the following assumption.

Assumption 10.2 (Identifiability Condition 2)**.** *For any* $0 \le k, l \le K$ *such that* $m_k \leqslant m_l$ *and* $n_k \leqslant n_l$*, we assume*

$$\operatorname{tr}\left[\boldsymbol{A}_l(\boldsymbol{A}_k \otimes \mathbf{1}_{i,j}^{m_l-m_k,n_l-n_k})'\right] = 0,$$

for all $1 \le i \le 2^{m_l-m_k}$ *and* $1 \le j \le 2^{n_l-n_k}$*, where* $\mathbf{1}_{i,j}^{m_l-m_k,n_l-n_k}$ *denotes the* $2^{m_l-m_k} \times 2^{n_l-n_k}$ *matrix whose* (i,j)*-th element is 1, and all other elements are 0.*

In particular, if $m_k = m_l$ and $n_k = n_l$, the condition reduces to

$$\operatorname{tr}[\boldsymbol{A}_l \boldsymbol{A}_k'] = \operatorname{tr}[\boldsymbol{B}_l \boldsymbol{B}_k'] = 0.$$

Therefore, Assumption 10.2 requires essentially the "orthogonality" of $\boldsymbol{A}_k$ and $\boldsymbol{A}_l$ when both dimensions of $\boldsymbol{A}_k$ are less than or equal to those of $\boldsymbol{A}_l$. If $\boldsymbol{A}_k$ and $\boldsymbol{A}_l$ do not satisfy the condition in Assumption 10.2, one can always perform an orthogonalization operation by finding a $2^{m_l-m_k} \times 2^{n_l-n_k}$ non-zero matrix $\boldsymbol{C}$, whose (i,j)-th element is given by

$$[\boldsymbol{C}]_{i,j} = \operatorname{tr}\left[\boldsymbol{A}_l(\boldsymbol{A}_k \otimes \mathbf{1}_{i,j}^{m_l-m_k,n_l-n_k})'\right], \tag{10.3}$$

such that Assumption 10.2 is satisfied for $\boldsymbol{A}_k$ and $\boldsymbol{A}_l^* = (\boldsymbol{A}_l - \boldsymbol{A}_k \otimes \boldsymbol{C})/\|\boldsymbol{A}_l - \boldsymbol{A}_k \otimes \boldsymbol{C}\|_F$. Note that $\boldsymbol{C}$ in (10.3) is the least squares solution of $\min \|\boldsymbol{A}_l - \boldsymbol{A}_k \otimes \boldsymbol{C}\|_F^2$. The procedure of orthogonalizing $\boldsymbol{A}_k$ and $\boldsymbol{A}_l$ can be generalized to multiple terms through the Gram-Schmidt process depicted in Algorithm 10. Note that the identifiability Assumption 10.2 can be replaced by the same condition on the $\boldsymbol{B}$'s, but there is no need to impose the condition on both $\boldsymbol{A}$'s and $\boldsymbol{B}$'s.

10.3 Estimation

In this section, we consider the estimation of $\boldsymbol{A}_k$ $\boldsymbol{B}_k$ and λ_k for the hKoPA model (10.1). When the configuration set $\mathcal{C}$ is known, we adopt a backfitting procedure (or alternating least squares) to fit the model. When the configurations are unknown, we propose a greedy algorithm by adding one Kronecker product component at a time.

Algorithm 10: Gram-Schmidt Process for hKoPA Model

Sort the configurations $\{(m_k, n_k)\}_{k=1}^K$ in ascending order such that (1) $m_i \leqslant m_j$ for all $i \leqslant j$; (2) $n_i \leqslant n_j$ if $m_i = m_j$;
Set $\boldsymbol{A}_1^* = \boldsymbol{A}_1$, $\boldsymbol{B}_1^* = \boldsymbol{B}_1$, $\lambda_1^* = \lambda_1$;
for $i = 2, \cdots, K$ **do**
  Let $\Omega_i = \{k < i : m_k \leqslant m_i, n_k \leqslant n_i\}$;
  $(\hat{\boldsymbol{C}}_k^i)_{k\in\Omega_j} = \underset{(\boldsymbol{C}_k^i)_{k\in\Omega_j}}{\arg\min} \left\| \boldsymbol{A}_i - \sum_{k\in\Omega_j} \boldsymbol{A}_k^* \otimes \boldsymbol{C}_k^i \right\|_F^2$;
  $\boldsymbol{R}_i = \boldsymbol{A}_i - \sum_{k\in\Omega_j} \boldsymbol{A}_k^* \otimes \hat{\boldsymbol{C}}_k^j$;
  $\boldsymbol{A}_i^* = \boldsymbol{R}_i / \|\boldsymbol{R}_i\|_F$, $\boldsymbol{B}_i^* = \boldsymbol{B}_i$;
  $\lambda_i^* = \lambda_i \|\boldsymbol{R}_i\|_F$;
  for $k \in \Omega_i$ **do**
    $\boldsymbol{S}_k = \boldsymbol{B}_k^* + \lambda_i \hat{\boldsymbol{C}}_k^i \otimes \boldsymbol{B}_k^* / \lambda_k^*$;
    $\boldsymbol{B}_k^* = \boldsymbol{S}_k / \|\boldsymbol{S}_k\|_F$;
    $\lambda_k^* = \lambda_k^* \|\boldsymbol{S}_k\|_F$;
  end
end
return $\{(\lambda_i^*, \boldsymbol{A}_i^*, \boldsymbol{B}_i^*)\}_{i=1}^K$.

10.3.1 Hybrid Kronecker Product Model with Known Configurations

When the configurations (m_k, n_k), $k = 1, \ldots, K$, are known, we consider the following least squares problem.

$$\min \left\| \boldsymbol{Y} - \sum_{k=1}^K \lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k \right\|_F^2. \tag{10.4}$$

When $K = 1$, such a problem can be solved by singular value decomposition of a rearranged version of matrix $\boldsymbol{Y}$ using the rearrangement operator $\mathcal{R}$ as shown in Section 8.3.

Therefore, the least squares optimization problem

$$\min \|\boldsymbol{Y} - \lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2,$$

is equivalent to a rank-one matrix approximation problem since

$$\|\boldsymbol{Y} - \lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2 = \|\mathcal{R}_{m,n}[\boldsymbol{Y}] - \lambda \mathrm{vec}(\boldsymbol{A})\mathrm{vec}(\boldsymbol{B})'\|_F^2,$$

whose solution is given by the leading component in the SVD of $\mathcal{R}_{m,n}[\boldsymbol{Y}]$ (Eckart and

Young, 1936).

When there are multiple terms $K > 1$ in model (10.1), we propose to solve the optimization problem (10.4) through a backfitting algorithm (or alternating least squares algorithm) by iteratively estimating λ_k, $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ by

$$\min \left\| \left(\boldsymbol{Y} - \sum_{i \neq k} \hat{\lambda}_i \hat{\boldsymbol{A}}_i \otimes \hat{\boldsymbol{B}}_i \right) - \lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k \right\|_F^2,$$

using the rearrangement operator and SVD, with fixed $\hat{\lambda}_i$, $\hat{\boldsymbol{A}}_i$ and $\hat{\boldsymbol{B}}_i$ $(i \neq k)$ from the previous iterations.

When all configurations $\{(m_k, n_k)\}_{k=1}^K$ are distinct, the backfitting procedure for hKoPA is depicted in Algorithm 11, where $\text{vec}_{m,n}^{-1}$ is the inverse of the vectorization operation that convert a column vector back to a $2^m \times 2^n$ matrix. When r terms indexed by $k_1, \ldots, k_r$ in the hKoPA model have the same configuration, these terms are updated simultaneously in the backfitting algorithm by keeping the first r components from the SVD of the residual matrix $\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{i \neq k_1, \ldots, k_r} \hat{\lambda}_i \hat{\boldsymbol{A}}_i \otimes \hat{\boldsymbol{B}}_i$. We also orthonormalize the components by the Gram-Schmidt procedure (Algorithm 10) at the end of each backfitting round.

Since each iteration of the backfitting procedure reduces the sum of squares of residuals, the algorithm always converges, though it may land in a local optimal. Empirical experiences show that most of the time the global minimum is reached. Starting with different initial values and with different orders of backfitting helps.

10.3.2 Hybrid KoPA with Unknown Configurations

In this section, we consider the case when the model configuration $\mathcal{C} = \{(m_k, n_k)\}_{k=1}^K$ is unknown. We use a greedy method similar to boosting to obtain the approximation by iteratively adding one Kronecker product at a time, based on the residual matrix obtained from the previous iteration. Specifically, at iteration k, we obtain

$$\hat{\boldsymbol{E}}^{(k)} = \boldsymbol{Y} - \sum_{i=1}^{k-1} \hat{\lambda}_i \hat{\boldsymbol{A}}_i \otimes \hat{\boldsymbol{B}}_i,$$

Algorithm 11: Backfitting Least Squares Procedure

Set $\hat{\lambda}_1 = \hat{\lambda}_2 = \cdots = \hat{\lambda}_K = 0$;
repeat
  for $k = 1, \cdots, K$ **do**
    $\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{i \neq k} \hat{\lambda}_i \hat{\boldsymbol{A}}_i \otimes \hat{\boldsymbol{B}}_i$;
    Compute SVD of $\mathcal{R}_{m_k, n_k}[\boldsymbol{E}_k]$:

$$\mathcal{R}_{m_k, n_k}[\boldsymbol{E}_k] = \sum_{j=1}^{J} s_j \boldsymbol{u}_j \boldsymbol{v}_j^T.$$

    Update $\hat{\lambda}_k = s_1$, $\hat{\boldsymbol{A}}_k = \mathrm{vec}^{-1}_{m_k, n_k}(\boldsymbol{u}_1)$ and $\hat{\boldsymbol{B}}_k = \mathrm{vec}^{-1}_{M-m_k, N-n_k}(\boldsymbol{v}_1)$;
  end
  Orthonormalize the components by Algorithm 10;
until *convergence*;
return $\{(\hat{\lambda}_k, \hat{\boldsymbol{A}}_k, \hat{\boldsymbol{B}}_k)\}_{k=1}^K$.

where $\hat{\lambda}_i$, $\hat{\boldsymbol{A}}_i$ and $\hat{\boldsymbol{B}}_i$ are obtained in the previous iterations, starting with $\boldsymbol{E}^{(1)} = \boldsymbol{Y}$. Then we use the single-term KoPA with unknown configuration proposed in Section 9.2.3 to obtain

$$\min_{\lambda_k, \boldsymbol{A}_k, \boldsymbol{B}_k} \|\hat{\boldsymbol{E}}^{(k)} - \lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k\|_F^2,$$

where the configuration (m_k, n_k) of $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{B}}_k$ is obtained by minimizing the information criterion

$$\mathrm{IC}_\kappa(m, n) = 2^{M+N} \ln \frac{\|\hat{\boldsymbol{E}}^{(k)} - \lambda_k \boldsymbol{A}_k \otimes \boldsymbol{B}_k\|_F^2}{2^{M+N}} + \kappa \eta, \tag{10.5}$$

where $\eta = 2^{m+n} + 2^{M+N-m-n}$ is the number of parameters of the single-term model with configuration (m_k, n_k) and q is the penalty coefficient on model complexity. As shown in Section 9.3, in a single-term Kronecker product case, when the signal-to-noise ratio is large enough, minimizing the information criterion IC_κ in (10.5) produces consistent estimators of the true configuration.

The procedure is repeated until a stopping criterion is reached as detailed in Algorithm 12.

The iterative Algorithm 12 is a greedy algorithm, which does not guarantee a global optimal in all configuration combinations. The output of the algorithm satisfies Assumption 10.1 but does not satisfy Assumption 10.2. However, searching the configuration space

Algorithm 12: Iterative Algorithm for hKoPA Estimation

Set $\hat{\boldsymbol{E}}^{(1)} = \boldsymbol{Y}$;
for $k = 1, \cdots, K$ **do**
  for *all possible configuration* (m, n) **do**
    Compute SVD for $\mathcal{R}_{m,n}[\hat{\boldsymbol{E}}^{(k)}]$:

$$\mathcal{R}_{m,n}[\hat{\boldsymbol{E}}^{(k)}] = \sum_{j=1}^{J} s_j u_j v_j'.$$

    Set $\hat{\lambda}_k^{(m,n)} = s_1$, $\hat{\boldsymbol{A}}_k^{(m,n)} = \mathrm{vec}_{m,n}^{-1}(u_1)$ and $\hat{\boldsymbol{B}}_k^{(m,n)} = \mathrm{vec}_{M-m,N-n}^{-1}(v_1)$;
    Compute $\hat{\boldsymbol{S}}_k^{(m,n)} = \hat{\lambda}_k^{(m,n)} \hat{\boldsymbol{A}}_k^{(m,n)} \otimes \hat{\boldsymbol{B}}_k^{(m,n)}$;
  end
  Compute

$$(\hat{m}_k, \hat{n}_k) = \underset{(m,n)}{\arg\min}\ 2^{M+N} \ln \frac{\|\hat{\boldsymbol{E}}^{(k)} - \hat{\boldsymbol{S}}_k^{(m,n)}\|_F^2}{2^{M+N}} + \kappa\eta.$$

  Set $\hat{\lambda}_k = \hat{\lambda}_k^{(\hat{m}_k, \hat{n}_k)}$, $\hat{\boldsymbol{A}}_k = \hat{\boldsymbol{A}}_k^{(\hat{m}_k, \hat{n}_k)}$ and $\hat{\boldsymbol{B}}_k = \hat{\boldsymbol{B}}_k^{(\hat{m}_k, \hat{n}_k)}$;
  if *a stopping criterion is met* **then**
    Break;
  end
  $\hat{\boldsymbol{E}}^{(k+1)} = \hat{\boldsymbol{E}}^{(k)} - \hat{\boldsymbol{S}}_k^{(\hat{m}_k, \hat{n}_k)}$;
end
return $\{(\hat{\lambda}_k, \hat{\boldsymbol{A}}_k, \hat{\boldsymbol{B}}_k)\}_{k=1}^{\hat{K}}$, where $\hat{K}$ is the number of terms determined by the stopping criterion.

$\{(m_k, n_k)\}_{k=1}^{K}$ in a greedy and additive way requires less computational cost. It is possible that, given the configuration $\hat{\mathcal{C}} = \{(\hat{m}_k, \hat{n}_k), k = 1, \ldots, \hat{K}\}$ obtained in the greedy algorithm, a refinement step can be engaged using the algorithm proposed in Section 3.1. If more computational resources are available, the refinement can be done at the end of each iteration k based on $\hat{\mathcal{C}}_k = \{(\hat{m}_i, \hat{n}_i), i = 1, \ldots, k\}$ to obtain better partial residual matrix $\hat{\boldsymbol{E}}^{(k)}$ so the configuration determination in future iterations are more accurate.

The stopping criterion can be selected according to the objective of the study. For denoising applications, one may specify the desired level of the proportion of the total variation explained by the hKoPA to be reached. We will introduce a practical stopping criterion based on the random matrix theory in the example section.

10.4 Simulation

In this simulation, we examine the performance of the least squares backfitting algorithm in Algorithm 11 for a two-term Kronecker product model and determine the factors that affect the estimation accuracy and convergence speed of the algorithm.

Specifically, we simulate the data matrix $\boldsymbol{Y}$ according to

$$\boldsymbol{Y} = \lambda_1 \boldsymbol{A}_1 \otimes \boldsymbol{B}_1 + \lambda_2 \boldsymbol{A}_2 \otimes \boldsymbol{B}_2 + \sigma \boldsymbol{E},$$

where $\lambda_1 = \lambda_2 = 1$, $\boldsymbol{A}_k$, $\boldsymbol{B}_k$ $(k = 1, 2)$ satisfy Assumption 10.1, and $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ satisfy Assumption 10.2. One of the objectives of the simulation study is to see the impact of linear dependence of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, since $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are already "linearly independent" under Assumption 10.2. The configurations for this simulation study are set to be

$$M = N = 9, \quad (m_1, n_1) = (4, 4), \quad (m_2, n_2) = (5, 5).$$

To simulate the component matrices, we first generate $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ with i.i.d. standard Gaussian entries, and then preform the Gram-Schmidt procedure given in Algorithm 1 so that they satisfy Assumption 10.2, and finally rescale them to have Frobenius norm one. We then generate $\tilde{\boldsymbol{B}}_1$ and $\tilde{\boldsymbol{B}}_2$ in exactly the same way (so that $\tilde{\boldsymbol{B}}_1$ and $\tilde{\boldsymbol{B}}_2$ also satisfy Assumption 10.2), and set

$$\boldsymbol{B}_1 = \frac{\tilde{\boldsymbol{B}}_1 + \alpha \mathbf{1} \otimes \tilde{\boldsymbol{B}}_2}{\sqrt{1 + \alpha^2 2^{m_2 + n_2 - m_1 - n_1}}}, \ \boldsymbol{B}_2 = \tilde{\boldsymbol{B}}_2,$$

where $\mathbf{1}$ is a $2^{m_2 - m_1} \times 2^{n_2 - n_1}$ matrix of ones.

It is seen that by such a construction α controls the linear dependency between $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$. In particular when $\alpha = 0$, $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are linearly independent in the sense that they satisfy Assumption 10.2. When $\alpha \to \infty$, $\boldsymbol{B}_1 \propto \mathbf{1} \otimes \boldsymbol{B}_2$ and the model can be represented using a single Kronecker product.

In this simulation, we consider $\alpha \in \{0, 0.5, 1.0, 1.5, 2.0\}$ and $\sigma_0 := 2^{(M+N)/2}\sigma \in \{0, 0.5, 1.0, 1.5, 2.0\}$. The benchmark setting is $\alpha = 0.5$ and $\sigma_0 = 1$, under which the signal-to-noise

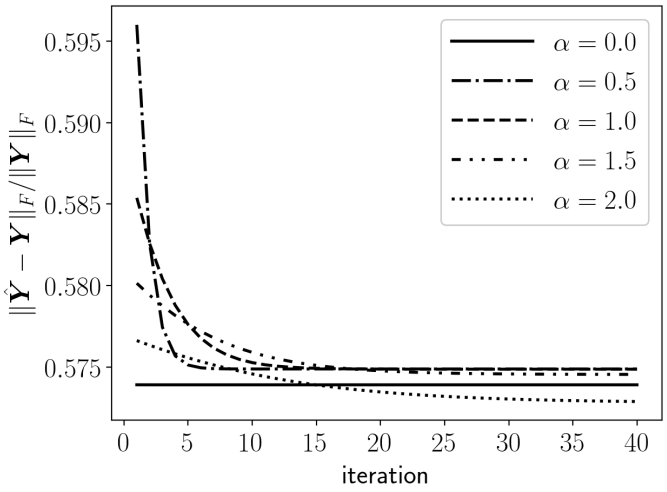


Figure 10.1: Fitting error against number of iterations for different α values, when $\sigma_0 = 1$

ratio is $(\lambda_1^2 + \lambda_2^2)/\sigma_0^2 = 2$.

We first examine the effect of linear dependency of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, controlled by α. Hence we fix $\sigma_0 = 1$ and check the performance of the backfitting algorithm under different values of α, using the known configurations. Figure 10.1 shows the relative error of $\hat{\boldsymbol{Y}}$ for the first 40 iterations for the five different values of α. At $\sigma_0 = 1$, a perfect fit is expected to have an error of

$$\frac{E[\|\sigma \boldsymbol{E}\|_F]}{E[\|\boldsymbol{Y}\|_F]} = \frac{\sigma_0}{\sqrt{\lambda_1^2 + \lambda_2^2 + \sigma_0^2}} \approx 0.577.$$

under the simulation setting. It is seen from Figure 10.1 that the estimators tend to overfit as the final relative errors are all smaller than the expected value. This is due to the fact that the observed error term $\boldsymbol{E}$ is not orthogonal to the observed signal. On the other hand, the less the linear dependence between $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, the less the model is overfitted.

By comparing the convergence speed of different α values, we notice that larger value of α, corresponding to higher linear dependency between $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, results in a slower convergence rate. When $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are linearly independent ($\alpha = 0$), only one iteration is needed.

In Figure 10.2, the errors of the estimators $\hat{\lambda}_k$, $\hat{\boldsymbol{A}}_k$, and $\hat{\boldsymbol{B}}_2$ are plotted against the number of iterations. For all the components, higher values of α result in slower convergence rates and less accurate final results. It is also seen that the estimation accuracy for $\boldsymbol{A}_k$ and

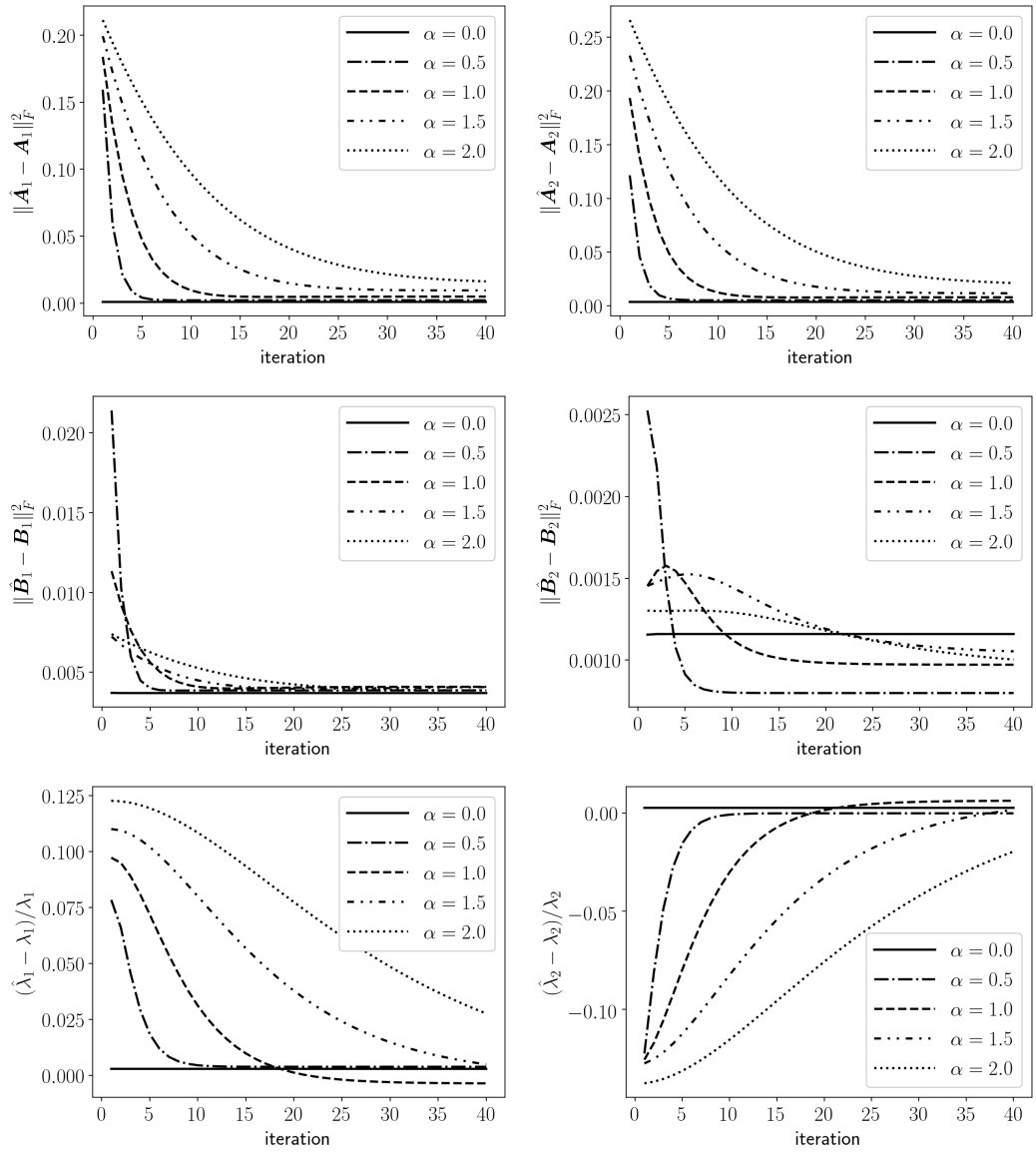


Figure 10.2: Errors against number of iterations at different α values for $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, λ_1 and λ_2.

$\boldsymbol{B}_k$ has quite different behaviors as the number of iterations increases. This is not surprising because $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are "linear independent", while $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are designed to have a linear dependency when $\alpha > 0$. In particular, the estimators $\hat{\boldsymbol{B}}_k$ seem to be more accurate than $\hat{\boldsymbol{A}}_k$.

Next, we examine the effect of the noise level σ_0. We fix $\alpha = 0.5$ and consider five different values of the noise σ_0. The error in estimating $\boldsymbol{Y}$ is reported in Figure 10.3. It is seen that higher noise level σ_0 results in larger errors, as expected. A small difference in the convergence speed is observed as well. The algorithm converges faster when the noise level is high, but it is not as sensitive as that in the change of linear dependence level α.

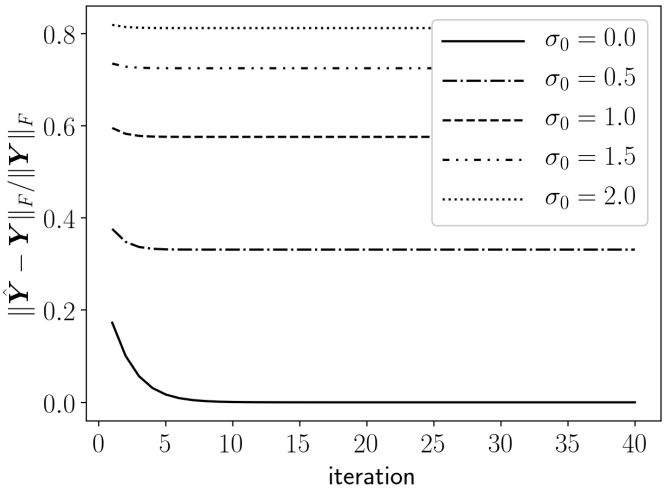


Figure 10.3: Fitting error against number of iterations for different σ_0 values

Errors for estimating the different components in the model are plotted in Figure 10.4 for different noise levels. The difference in convergence rates is less obvious. We also observe that the performance for estimating the smaller component matrices $\boldsymbol{A}_1$ and $\boldsymbol{B}_2$ is better than that for the larger matrices $\boldsymbol{A}_2$ and $\boldsymbol{B}_1$.

10.5 Example

In this section, we apply the hKoPA to analyze the image of Lenna, which has been used widely as a benchmark example in image processing. The Lenna image shown in Figure 10.5 is a gray-scaled 512×512 picture, which is represented by a 512×512 ($M = N = 9$) real

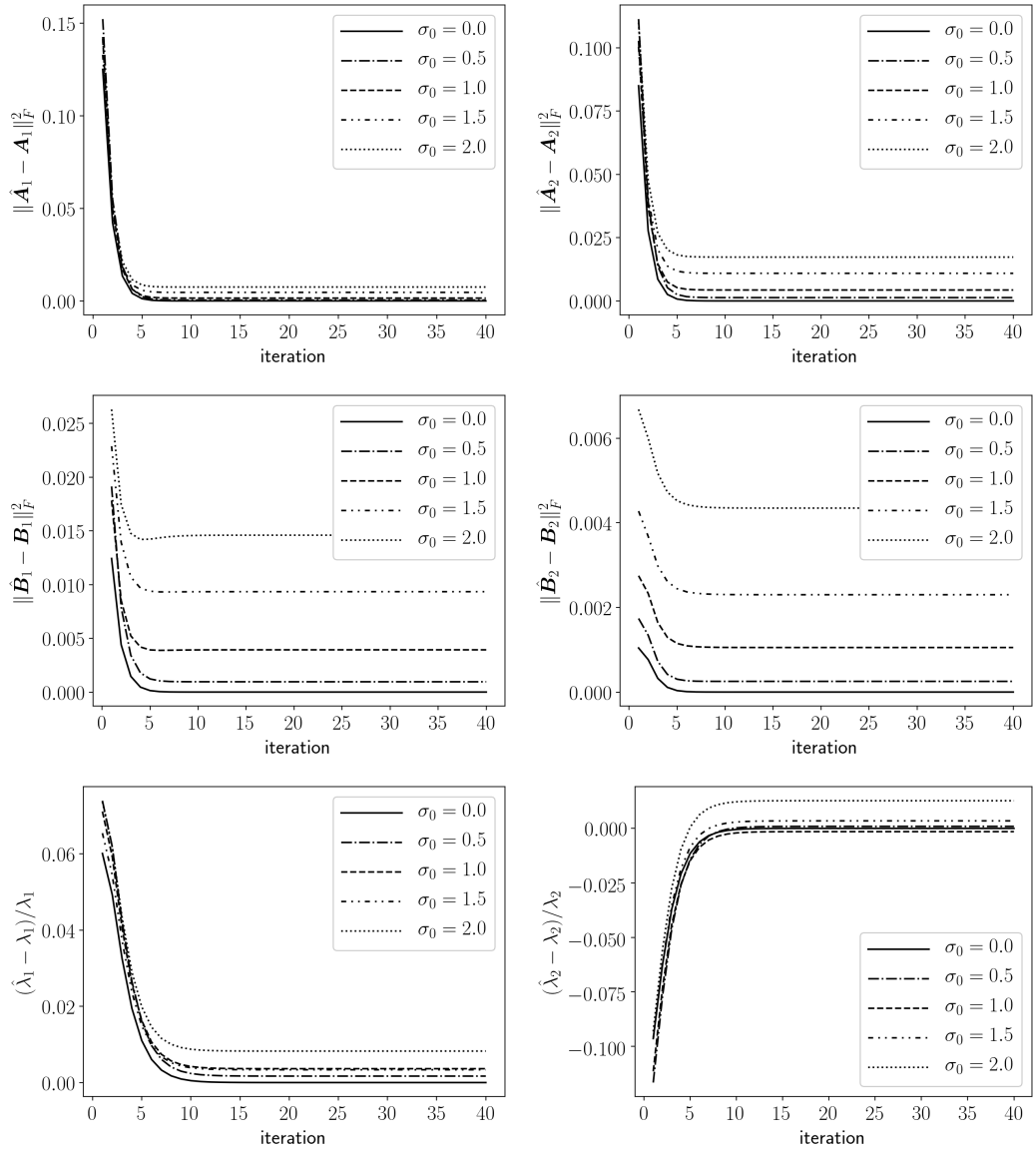


Figure 10.4: Errors in components against number of iterations for different σ_0 values.

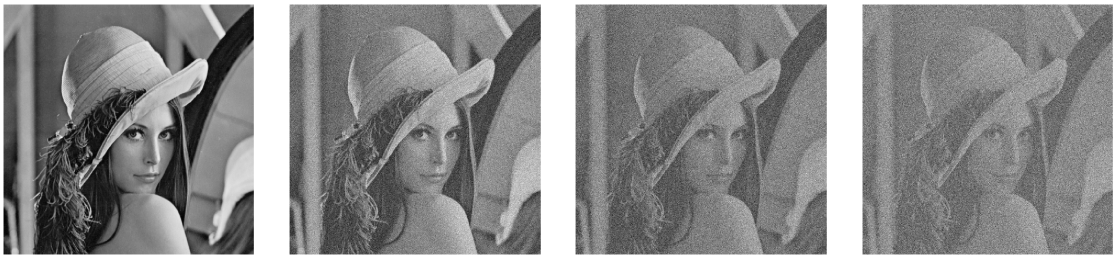


Figure 10.5: (Left) Original Grayscaled Lenna's image. (Mid Left) Noisy image with $\sigma = 0.1$. (Mid Right) Noisy image with $\sigma = 0.2$. (Right) Noisy image with $\sigma = 0.3$.

matrix $\boldsymbol{Y}_0$. The elements of $\boldsymbol{Y}_0$ are real numbers between 0 and 1, where 0 represents black and 1 represents white. Besides the original image, in this example we also consider some artificially corrupted images using

$$\boldsymbol{Y} = \boldsymbol{Y}_0 + \sigma \boldsymbol{E},$$

where $\boldsymbol{E}$ is a matrix of i.i.d. standard Gaussian random variables and σ denotes the noise level. We consider three noise levels $\sigma \in \{0.1, 0.2, 0.3\}$. Note that the original image scale is $[0, 1]$. Hence the image with noise level $\sigma = 0.3$ is considered to be heavily corrupted. The noisy images are shown in Figure 10.5.

For this example, the configurations in the hKoPA model (10.1) are unknown. Therefore, we adopt the iterative greedy algorithm proposed in Section 10.3.2, where the configuration in each iteration is determined by BIC. For each σ, we consider to fit the image with at most 20 Kronecker product terms. The selected configurations $(\hat{m}_k, \hat{n}_k)$, the estimated $\hat{\lambda}_k$ and the cumulative percentage of variation explained (c.p.v.) by the first 10 iterations are reported in Table 10.6. It is seen that for all noise levels σ, the first several Kronecker product terms can explain most of the variation of $\boldsymbol{Y}$. To check the possible overfitting, we report the ratio $\|\boldsymbol{Y}_0\|_F^2 / \|\boldsymbol{Y}\|_F^2$ in percentage at the bottom row of Table 10.6. When $\sigma = 0.3$, the c.p.v. exceeds this ratio after the seventh iteration, indicating the overfitting if more terms are added to hKoPA.

In the heavily corrupted cases, configurations close to the center such as $(5, 4)$ are more likely to be selected by BIC. These configurations correspond to more squared $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ matrices.

k	$\sigma = 0.0$			$\sigma = 0.1$			$\sigma = 0.2$			$\sigma = 0.3$		
	$(\hat{m}_k, \hat{n}_k)$	$\hat{\lambda}_k$	c.p.v.	$(\hat{m}_k, \hat{n}_k)$	$\hat{\lambda}_k$	c.p.v.	$(\hat{m}_k, \hat{n}_k)$	$\hat{\lambda}_k$	c.p.v.	$(\hat{m}_k, \hat{n}_k)$	$\hat{\lambda}_k$	c.p.v.
1	(6, 7)	95.07	91.81	(5, 6)	91.21	66.76	(5, 6)	91.80	41.40	(4, 6)	88.28	23.21
2	(6, 6)	14.21	93.86	(5, 6)	21.88	70.60	(3, 6)	15.42	42.57	(4, 5)	26.97	25.37
3	(5, 7)	12.18	95.39	(5, 6)	19.48	73.65	(5, 4)	14.07	43.58	(4, 5)	18.05	26.34
4	(6, 6)	10.17	96.47	(4, 5)	8.00	74.16	(5, 4)	13.52	44.47	(3, 6)	17.37	27.24
5	(5, 6)	6.47	96.90	(5, 4)	7.66	74.63	(5, 4)	12.60	45.25	(4, 5)	15.68	27.97
6	(5, 5)	4.65	97.12	(4, 5)	7.00	75.03	(3, 6)	11.91	45.96	(4, 5)	15.24	28.66
7	(4, 5)	3.48	97.24	(4, 5)	6.67	75.39	(3, 6)	11.07	46.60	(3, 6)	14.70	29.32
8	(4, 5)	3.29	97.35	(5, 4)	6.40	75.72	(5, 4)	10.56	47.13	(4, 5)	13.84	29.90
9	(5, 5)	3.66	97.49	(5, 4)	6.19	76.03	(5, 4)	9.92	47.62	(4, 5)	13.76	30.47
10	(4, 5)	2.92	97.58	(5, 4)	6.05	76.32	(3, 6)	9.59	48.08	(2, 7)	13.63	31.00
$\boldsymbol{Y}$	-	-	100	-	-	79.01	-	-	48.38	-	-	29.32

Figure 10.6: The selected configurations, $\hat{\lambda}_k$, and the cumulative percentage of variation (c.p.v.) explained by the first 10 iterations. The bottom row gives $\|\boldsymbol{Y}_0\|_F^2/\|\boldsymbol{Y}\|_F^2$ in percentage.

Figure 10.7: Fitted images in the first, third and fifth iterations. (Row 1) $\sigma = 0.0$. (Row 2) $\sigma = 0.1$. (Row 3) $\sigma = 0.2$. (Row 4) $\sigma = 0.3$.

The recovered images using one to five Kronecker product terms at different noise levels σ are given in Figure 10.7, where the total number of parameters involved is shown under each image. We see that hKoPA is able to recovered the true image with a small number of iterations. Even in the most noisy case $\sigma = 0.3$, lots of details are present.

In addition to the iterative greedy algorithm, we propose a stopping criterion based on random matrix theory to determine the number of Kronecker products. Specifically, at iteration k, an estimate of σ is

$$\hat{\sigma} = 2^{-(M+N)/2} \|\hat{\boldsymbol{E}}^{(k)} - \hat{\boldsymbol{S}}_k\|_F = 2^{-(M+N)/2} \|\hat{\boldsymbol{E}}^{(k+1)}\|_F.$$

Under the i.i.d. Gaussian assumption on $\hat{\boldsymbol{E}}^{(k+1)}$, we have

$$P\left[\|\mathcal{R}_{\hat{m}_k,\hat{n}_k}[\hat{\boldsymbol{E}}_{k+1}]\|_S \geqslant \hat{\sigma}\left(2^{(\hat{m}_k+\hat{n}_k)/2} + 2^{(M+N-\hat{m}_k-\hat{n}_k)/2} + t\right)\right] \leqslant e^{-t^2/2},$$

according to the non-asymptotic analysis on the random matrices and the concentration inequalities (Vershynin, 2010). Here we set $t = \sqrt{2\log 100} \approx 3.03$ such that the probability is bounded by 0.01. We terminate the algorithm at step k if

$$\frac{\hat{\lambda}_k}{\hat{\sigma}} \leqslant 2^{(\hat{m}_k+\hat{n}_k)/2} + 2^{(M+N-\hat{m}_k-\hat{n}_k)/2} + \sqrt{2\log 100}, \tag{10.6}$$

and use the first $\hat{K} = k-1$ terms as the optimal approximation. Specifically, when $\sigma = 0$ or 0.1, the stopping criterion is never met in the first 20 iterations and we use $\hat{K} = 20$. When $\sigma = 0.2$, a 9-term model is selected, and when $\sigma = 0.3$, the stopping criterion results in a 7-term model.

Low rank approximation is another widely used approach in image denoising and compression. It assumes

$$\boldsymbol{Y} = \sum_{i=1}^{K} \lambda_i u_i v_i'.$$

The complexity is controlled by the number of rank one matrices K. We remark that the low rank approximation is a special case of hKoPA. It corresponds to the case that all Kronecker products in (10.1) are of the same configuration $(M, 1)$. To compare the performance of

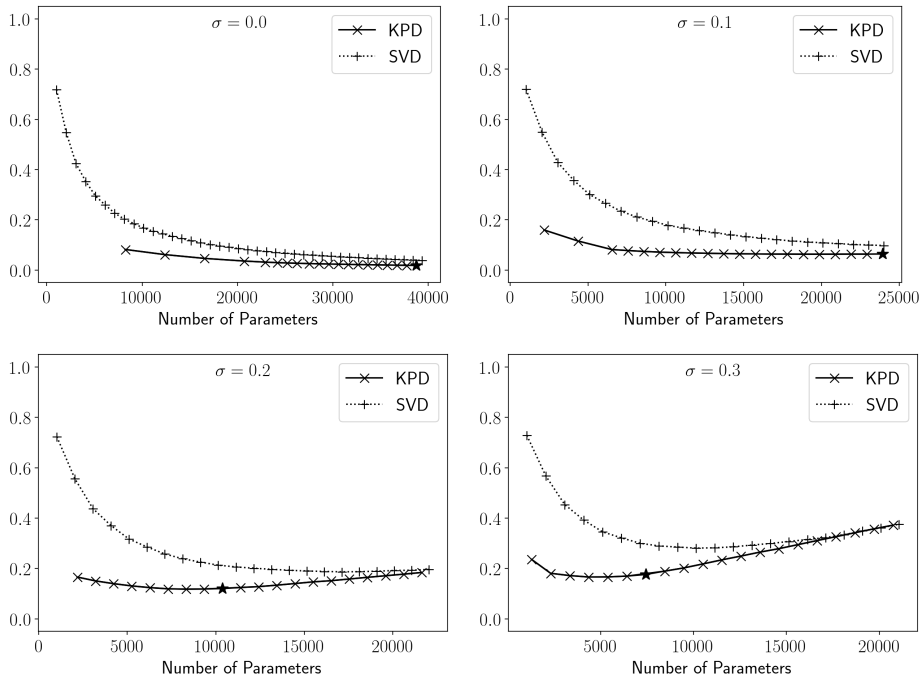


Figure 10.8: RSE error of against the number of parameters used for KPD and SVD approaches at different noise levels. The optimal model determined by empirical stopping rule is marked by '⋆'.

hKoPA with the low rank approximation approach, we calculate the relative squared error of the fitted matrix $\hat{\boldsymbol{Y}}$ by

$$RSE = \frac{\|\boldsymbol{Y}_0 - \hat{\boldsymbol{Y}}\|_F^2}{\|\boldsymbol{Y}_0\|_F^2},$$

where $\boldsymbol{Y}_0$ is the original image without noise and $\hat{\boldsymbol{Y}}$ is the fitted matrix of the noisy version $\boldsymbol{Y}$. For both hKoPA and the low rank approximation, and for different noise levels σ, we plot RSE against the number of parameters used in the approximation, in Figure 10.8. For each graph, the hKoPA chosen by the proposed stopping criterion (10.6) is marked with a "⋆". Comparing the error curve of SVD with the one of hKoPA, Figure 10.8 reveals that for any level of model complexity (or the number of parameters), hKoPA is more accurate than the standard low rank SVD approximation. When noises are added, overfitting is observed for both hKoPA and SVD approximation as the error (compared to the true image) increases when too many terms are used, as seen from the U-shape of the curves. The stopping criterion in (10.6) prevents the model from significantly overfitting. The realized relative error of the hKoPA with the number of terms selected by (10.6) is close to the minimum

attainable error, though the estimated number of terms is not the optimal one.

CHAPTER 11

Matrix Completion with KoPA

11.1 Introduction to Matrix Completion Problems

Many applications involve observations in a high dimensional matrix form. Often the observed matrix has a certain number of missing entries and is observed with error. In many recent machine learning studies of high dimensional matrix observations, a common approach is to assume that the observed matrix has a underlying low rank structure. The low rank representation explains the interaction between matrix entries with a smaller number of parameters and reveals the core factors that drive and control the high dimensional observations, resulting in significant dimension reduction. Such a low rank assumption also makes it possible to recover the missing entries in an observed matrix, which is known as the matrix completion problem. Some of the matrix completion applications include collaborative filtering (Goldberg et al., 1992), global positioning (Biswas et al., 2006) and remote sensing (Schmidt, 1986). One of the most famous examples is the Netflix recommendation system contest, in which the winning algorithm recovers the movie-rating matrix by a rank one matrix based on the observed entries.

Two different settings of matrix completion problems have been studied in the literature. One is the exact matrix completion problem whose goal is to recover the original matrix *exactly* when a portion of the matrix entries is missing. When the original matrix rank is known, it can be recovered through the alternating minimization algorithm proposed by Jain et al. (2013) under certain conditions. When the matrix rank is unknown, it is still possible to exactly recover the matrix through nuclear norm optimization (Candès and

Recht, 2009; Candès and Tao, 2010). The nuclear norm optimization approach can also be applied to tensor completion problems whose goal is to recover a tensor structure (Yuan and Zhang, 2016). The second setting considers the circumstances when the observed entries are corrupted with noises while a portion of the matrix entries is missing. It is known as the stable matrix completion problem. Candes and Plan (2010) extends the nuclear norm optimization approach in the exact matrix completion problem to the stable matrix completion problem by relaxing the constraint. Assuming the matrix rank is known, Keshavan et al. (2010) approaches the problem using a combination of spectral techniques and manifold optimization. Specifically for stable rank one matrix completion problem, Cosse and Demanet (2017) proposes to solve it using two rounds of semi-definite programming relaxation. Note that the alternating minimization algorithm in Jain et al. (2013) is applicable for the stable matrix completion problem as well.

It is observed that in many applications of image processing, signal processing and quantum computing, the high dimensional data in matrix forms often has a low-rank structure in terms of Kronecker product instead of singular value decomposition (Werner et al., 2008; Duarte and Baraniuk, 2012; Kamm and Nagy, 1998). Approximating a matrix with a sum of a small number of matrices in Kronecker product form is an extension of low rank approximation with a sum of rank one matrices. The flexibility provides an alternative approach for matrix completion. The key challenging factor of the approach is to determine the configurations of the Kronecker product.

In this chapter, we consider the matrix completion problem under the setting that the original matrix has a k-term Kronecker product structure with an unknown configuration. We propose to use an information criterion to determine the configuration similar to the one in Section 9.2.3. Particularly we first evaluate the information criterion of each possible conformable configurations, then the matrix is completed under the Kronecker product structure, using the chosen configuration.

11.2 Matrix Completion with KoPA

11.2.1 Matrix Completion

Let $\boldsymbol{X} \in \mathbb{R}^{P\times Q}$ be a high dimensional matrix. Suppose both P and Q can be factorized as $P = pp^*$ and $Q = qq^*$. In opposite to the complete Kronecker Product Decomposition in Section 8.2, a low rank KPD assumes that

$$\boldsymbol{X} = \sum_{i=1}^{r} \lambda_i \boldsymbol{A}_i \otimes \boldsymbol{B}_i \tag{11.1}$$

with $r < (pq) \wedge (p^*q^*)$. This is similar to matrix low rank assumption in which the matrix $\boldsymbol{X}$ is assumed to have the form

$$\boldsymbol{X} = \sum_{i=1}^{r} \lambda_i u_i v_i', \tag{11.2}$$

a sum of r rank one matrices. In fact, there is a direct connection between (11.1)) and (11.2) as explained later.

Let $\boldsymbol{Y}^*$ be the observed matrix with missing entries to be estimated. We assume

$$[\boldsymbol{Y}^*]_{ij} = [\boldsymbol{Y}]_{ij}\delta_{ij}$$

where δ_{ij} i.i.d $\sim$ Bernoulli(τ), and $\boldsymbol{Y}$ is the complete data matrix. The rate τ is called the observing rate. We further assume $\boldsymbol{Y}$ is a corrupted version of an underlying signal matrix $\boldsymbol{X}$ in that

$$\boldsymbol{Y} = \boldsymbol{X} + \sigma\boldsymbol{E}, \tag{11.3}$$

where $\boldsymbol{E}$ is a $P \times Q$ matrix with i.i.d. standard Gaussian entries and σ denotes the noise level. The underlying signal matrix $\boldsymbol{X}$ is assumed to have a low rank KPD in (11.1).

Let Ω be the set of the indices of observed entries in $\boldsymbol{Y}^*$ and define $\boldsymbol{Y}^* = P_\Omega(\boldsymbol{Y})$ be the observed version of $\boldsymbol{Y}$ such that

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

Note that, the standard matrix completion algorithms (Candès and Recht, 2009; Candes and Plan, 2010; Jain et al., 2013) assume a low rank structure of the signal matrix $\boldsymbol{X}$ in the form of (11.2). It is a special case of our model, with $p = P, p^* = 1$ and $q = 1, q^* = Q$. In this paper we propose to use an information criterion to determine the configuration (p, q) and the rank r as well, for the purpose of matrix completion under the low rank KPD setting.

We propose a two-step procedure to solve the stable matrix completion problem. In the first step, we explore all candidate configurations of the underlying KPD. For each configuration, the Kronecker matrix completion problem is equivalent to the classical matrix completion problem after a rearrangement operation of the matrix elements and hence is solved with existing low rank matrix completion algorithms. We obtain an information criterion for each possible configurations. In the second step, we determine the configuration by minimizing the information criterion over all candidate configurations and the final completion estimate is obtained using the estimated optimal configuration.

11.2.2 Estimation

As discussed in Section 8.3, the Kronecker product of two matrices and the outer product of their vectorized version are linked through a rearrangement operation such that

$$\mathcal{R}_{p,q}[\boldsymbol{A} \otimes \boldsymbol{B}] = \mathrm{vec}(\boldsymbol{A})[\mathrm{vec}(\boldsymbol{B})]'.$$

That is, for any given configuration, the rearrangement operation $\mathcal{R}$ turns a Kronecker product into a rank-one matrix.

Let (p, q) be any allowed configuration for a $P \times Q$ matrix. That is, p is a factor of P and q is a factor of Q. Let r be a fixed rank. Given any partially observed corrupted matrix $P_\Omega(\boldsymbol{Y})$ of dimensions P and Q, the Kronecker matrix completion problem under configuration (p, q) is the following optimization problem.

$$\min_{\lambda_i, \boldsymbol{A}_i, \boldsymbol{B}_i} \left\| P_\Omega \left(\sum_{i=1}^{r} \lambda_i \boldsymbol{A}_i \otimes \boldsymbol{B}_i \right) - P_\Omega \boldsymbol{Y} \right\|_F . \tag{11.4}$$

Since rearrangement operation preserves the Frobenius norm, the optimization problem

Algorithm 13: Matrix Completion with Fixed Configuration

Input: matrix $P_\Omega \boldsymbol{Y}$, configuration (p, q), rank r;
Let $\tilde{\boldsymbol{Y}} = P_{\bar{\Omega}_{p,q}} \mathcal{R}_{p,q}[\boldsymbol{Y}]$;
Initialize $\boldsymbol{\Lambda}^{(0)}$, $\boldsymbol{U}^{(0)}$, $\boldsymbol{V}^{(0)}$ such that $\boldsymbol{U}^{(0)}\boldsymbol{\Lambda}^{(0)}[\boldsymbol{V}^{(0)}]'$ is the leading rank r SVD of $\tilde{\boldsymbol{Y}}$;
repeat
  $\boldsymbol{U}^* = \arg\min_{\boldsymbol{U}} \|P_{\bar{\Omega}_{p,q}}[\tilde{\boldsymbol{Y}} - \boldsymbol{U}[\boldsymbol{V}^{(k)}]']\|_F$;
  $\boldsymbol{V}^* = \arg\min_{\boldsymbol{V}} \|P_{\bar{\Omega}_{p,q}}[\tilde{\boldsymbol{Y}} - \boldsymbol{U}^*[\boldsymbol{V}]']\|_F$;
  Update $\boldsymbol{\Lambda}^{(k+1)}$, $\boldsymbol{U}^{(k+1)}$, $\boldsymbol{V}^{(k+1)}$ such that $\boldsymbol{U}^*[\boldsymbol{V}^*]' = \boldsymbol{U}^{(k+1)}\boldsymbol{\Lambda}^{(k+1)}[\boldsymbol{V}^{(k+1)}]'$ is in standard SVD form;
until *convergence*;
return $\hat{\boldsymbol{X}} = \mathcal{R}^{-1}_{p,q}[\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}']$ where $\boldsymbol{\Lambda}, \boldsymbol{U}, \boldsymbol{V}$ are from the last iteration.

(11.4) is equivalent to the classical rank-r matrix completion problem

$$\min_{\lambda_i, u_i, v_i} \left\| P_{\bar{\Omega}_{p,q}} \left(\sum_{i=1}^{r} \lambda_i u_i v_i' - \mathcal{R}_{p,q}[\boldsymbol{Y}] \right) \right\|_F . \tag{11.5}$$

where $u_i = \mathrm{vec}(\boldsymbol{A}_i)$, $v_i = \mathrm{vec}(\boldsymbol{B}_i)$, and $\bar{\Omega}_{p,q}$ records the indices of observed entries after the rearrangement such that for any $pp^* \times qq^*$ matrix $\boldsymbol{M}$, we have $\mathcal{R}_{p,q}[P_\Omega \boldsymbol{M}] = P_{\bar{\Omega}_{p,q}} \mathcal{R}_{p,q}[\boldsymbol{M}]$. To solve the optimization in (11.5), we adopt the alternating minimization algorithm proposed by Jain et al. (2013), where the initial values for u_i and v_i are directly estimated from the singular value decomposition of $P_{\bar{\Omega}_{p,q}} \mathcal{R}_{p,q}[\boldsymbol{Y}]$ as in Keshavan et al. (2010). The algorithm is depicted in Algorithm 13. The recovered matrix is therefore $\hat{\boldsymbol{X}} = \mathcal{R}^{-1}_{p,q}[\sum_{i=1}^{r} \hat{\lambda}_i \hat{u}_i \hat{v}_i']$, where $\mathcal{R}^{-1}$ is the inverse operation of rearrangement and $\hat{\lambda}_i$, $\hat{u}_i$ and $\hat{v}_i$ are the optimal solution of (11.5).

Notice that if one complete row or column of $\mathcal{R}_{p_1,q_1}[\boldsymbol{Y}]$ is missing, the matrix cannot be completely recovered by Algorithm 13. For example, if row j of $\mathcal{R}_{p.q}[\boldsymbol{Y}]$ is completely missing (corresponding to a missing block of size $P/p \times Q/q$ in $\boldsymbol{Y}$), the j-th value of u_i $(i = 1, \ldots, r)$ is not recoverable as it can have an arbitrary value and the value of the objective function in (11.5) does not change. In this case, the same block in the recovered matrix $\hat{\boldsymbol{X}} = \mathcal{R}^{-1}_{p,q}[\sum_{i=1}^{r} \hat{\lambda}_i \hat{u}_i \hat{v}_i']$ would remain to be missing. In classical matrix completion studies (Candès and Recht, 2009; Candes and Plan, 2010), a *feasibility condition* is often assumed to control the probability of missing complete row or column, which in turn imposes a condition

on the missing rate in relationship to the dimensions of the matrices (Chen et al., 2019b). In Kronecker product approach, we will provide two possible solutions in Section 11.3.

Another assumption often used in traditional low rank matrix completion algorithms is the *incoherence* condition, which assumes that the information of u_i and v_i spreads across all the columns and rows of the matrix $u_i v_i^T$. In the rank-r case as in (11.5), the incoherence condition bounds the ℓ_∞ norm of all u_i and v_i. In our case, the condition bounds the ℓ_∞ norm of all $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$.

We also note that most of the existing theoretical results (Candes and Plan, 2010; Chen et al., 2019b) for the optimization problem of (6) also apply for the optimization problem of (5). Hence with a known true configuration and the incoherence condition, these results hold for matrix completion algorithm under Kronecker product approximation.

11.2.3 Information Criterion for Configuration Determination

Section 11.2.2 provides the algorithm to estimate $\boldsymbol{X}$ from the partially observed $P_\Omega \boldsymbol{Y}$ when an arbitrary allowed Kronecker product configuration (p,q) is given. It remains to estimate the true configuration from all possible ones. Again, let (P,Q) be the dimension of $\boldsymbol{Y}^*$ and let $1 = p_0 < p_1 < \ldots < p_M = P$ be all the factors of P and $1 = q_0 < q_1 < \ldots < q_N = Q$ be all the factors of Q. Note that $p_m p_{M-m} = P$ and $q_n q_{N-n} = Q$ for all $0 \le m \le M$ and $0 \le n \le N$. To simplify notation, in the following, we use (m,n) to denote the configuration, instead of (p_m, q_n). All possible configurations are in the set

$$\mathcal{C} = \{(m,n) : m \in [M],\ n \in [N]\} \backslash \{(0,0),(M,N)\},$$

where the configurations $(0,0)$ and (M,N) are excluded because they correspond to trivial Kronecker products in which one of the matrices is a scalar.

We propose to determine the configuration through minimizing an information criterion,

$$IC_\kappa(m,n) = PQ \ln \frac{\|P_\Omega \hat{\boldsymbol{X}} - P_\Omega \boldsymbol{Y}\|_F^2}{|\Omega|} + \kappa \eta, \tag{11.6}$$

where $\hat{\boldsymbol{X}}$ is the recovered matrix under configuration (m,n), $\eta = p_m q_n + p_{M-m} q_{N-n}$ is the

number of parameters in the Kronecker product model (11.3) and κ is the coefficient of penalty on model complexity.

Here we heuristically discuss why the minimization in (11.6) gives the true configuration. For a fixed observing probability τ, when the size of matrix $\boldsymbol{Y}$ is large enough such that $PQ \to \infty$, the number of observed entries $|\Omega| \approx \tau PQ$ increases to infinity as well. Therefore, $\|P_\Omega \hat{\boldsymbol{X}} - P_\Omega \boldsymbol{Y}\|_F^2/|\Omega| \approx \|\hat{\boldsymbol{X}} - \boldsymbol{Y}\|_F^2/(PQ)$ since the elements in $\boldsymbol{Y}$ are observed independently. As pointed out in Section 11.2.2, $\hat{\boldsymbol{X}}$ is approximately the (Kronecker) low rank approximation of $\boldsymbol{Y}$ under configuration (m,n). The information criterion in (11.6) approximates the information criterion at $\tau = 1$, which is

$$IC_\kappa(m,n) = PQ \ln \frac{\|\boldsymbol{Y}\|_F^2 - \|\mathcal{R}_{m,n}[\boldsymbol{Y}]\|_S^2}{PQ} + \kappa\eta.$$

As proved in Section 9.3, the information criterion IC_κ can select the true configuration consistently when the signal-to-noise ratio $\|\boldsymbol{X}\|_F^2/(PQ\sigma^2)$ exceeds certain threshold. Consequently, the true configuration is expected to be selected consistently by the minimization in (11.8). A rigorous theoretical investigation is needed though.

11.3 Feasibility and Model Average

As discussed in Section 11.2.2, if a configuration produces a missing row or column after rearrangement, the recovered matrix $\hat{\boldsymbol{X}}$ by Algorithm 11 will still have unrecovered missing entries. One simple solution is to restrict the candidate configuration set to the *feasible configuration set*

$$\mathcal{C}_0 = \{(m,n) \in \mathcal{C} : P_{\bar{\Omega}_{m,n}} \mathcal{R}_{m,n}[\boldsymbol{Y}] \text{ has no missing column/row } \} \tag{11.7}$$

We may assume that the true configuration (m_*, n_*) belongs to $\mathcal{C}_0$. In other words, the Kronecker matrix completion problem defined in Section 11.2.1 is feasible at least for the true configuration. Then an estimator of the configuration can be obtained by

$$(\hat{m}, \hat{n}) = \underset{(m,n)\in\mathcal{C}_0}{\arg\min}\, IC_\kappa(m,n). \tag{11.8}$$

In Kronecker matrix completion, configurations close to the corners $(0,0)$ and (M,N) are more likely to be excluded from $\mathcal{C}_0$. To see it, recall τ is the probability that a element is observed. Hence the probability that at least one entire column in $\mathcal{R}_{m,n}[\boldsymbol{Y}]$ is missing is

$$P_c(m,n;M,N,\tau) = 1-[1-(1-\tau)^{p_m q_n}]^{p_{M-m}q_{N-n}}.$$

Similarly, the probability of missing an entire row in $\mathcal{R}_{m,n}[\boldsymbol{Y}]$ is

$$P_r(m,n;M,N,\tau) = P_c(M-m,N-n;M,N,\tau).$$

Assuming sufficiently large $M+N$, the configurations close to $(0,0)$ have $P_c(m,n;M,N,\tau) \approx 1$ and the configurations close to (M,N) have $P_r(m,n;M,N,\tau) \approx 1$. Figure 11.1 plots the probability P_c as a function of $m+n$ when $M+N=20$ for different values of τ, with $P=2^M$, $Q=2^N$, $p_m=2^m$ and $q_n=2^n$.

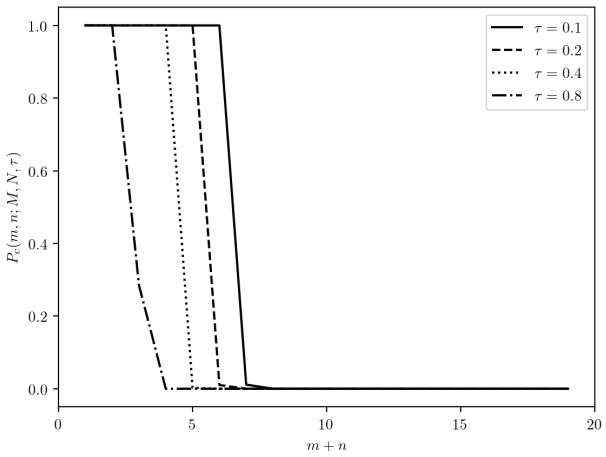


Figure 11.1: Function value of $P_c(m,n;M,N,\tau)$ when $M+N=20$ under the $P=2^M$ and $Q=2^N$ setting.

Intuitively it is easy to see the corner configurations (when pq or p^*q^* are small) have a higher probability to be infeasible. The rearranged matrix is of dimension $pq \times p^*q^*$. When pq or p^*q^* is small, the rearrangement matrix is either a short and fat matrix or a tall and thin matrix, hence it is easier to have a complete missing row or column. A different way to see the impact of corner configuration is through the number of infeasible entries under the

configuration. It is easily seen that the expected number of rows and columns missing in the reconfigured matrix is $p^*q^*(1-\tau)^{pq}$ and $pq(1-\tau)^{p^*q^*}$, respectively. Hence the proportion of infeasible entries (the missing entries after Kronecker production completion in Algorithm 1) is roughly $(1-\tau)^{(pq)\wedge p^*q^*}$.

In fact these corner configurations are less interesting. For example, if P is an even number, then the configuration $(p=2,q=1)$ uses a 2×1 vector as $\boldsymbol{A}_i$ and $P/2\times Q$ matrix as $\boldsymbol{B}_i$. Their Kronecker product is a matrix with its top and bottom halves differs by a multiplicative constant. And it uses approximately $PQ/2$ parameters, which would easily result in overfitting of the matrix. In additional, the theory in Section 9.3 indicates that the information criterion (under full matrix setting) requires higher signal-to-noise ratio in order to distinguish the true configuration and the corner cases. In the matrix completion problem this issue is more severe since the first term in the information criterion (11.6) is based only on the observed observations. In addition, the Kronecker rank r cannot be larger than the minimum of pq and p^*q^* based on the complete Kronecker decomposition in Section 8.2. Hence the corner configurations limits the number of Kronecker product terms in the model. Hence it is often beneficial to exclude the corner cases from consideration.

Instead of just simply assuming the true configuration is in the feasible set, an alternative is to combine several configurations in a model averaging operation. Since different configurations rearrange the observed matrix differently, the infeasible entry sets (the entries that cannot be recovered under a configuration) may not overlap for different configurations. Hence the infeasible entries under a configuration may be estimated with an alternative configuration that is feasible for these specific entries. We propose the following model averaging operation.

First we define the restricted configuration set as

$$\mathcal{C}_s = \{(m,n)\in\mathcal{C} : p_m q_n \wedge p_{M-m} q_{N-n} \geq \ln(s)/\ln(1-\tau)\}, \tag{11.9}$$

where s controls the expected proportion of infeasible entries under a configuration.

Let $C_k = (p_{m_k}, q_{n_k}), k=1,2,\ldots$ be the sequence of configurations ordered according to their information criterion $\mathrm{IC}(C_k)$ within the restricted configuration set $\mathcal{C}_s$, and let $\hat{\boldsymbol{X}}_k$ be

the estimated $\boldsymbol{X}$ using configuration C_k. Define ν_{ijk} be the infeasible entry indicator for which $\nu_{ijk} = 0$ if (i,j)-th entry is an infeasible entry under configuration C_k, and $\nu_{ijk} = 1$ otherwise. Let $d_{ij} = \min_k\{\nu_{ijk} = 1\}$ so $C_{d_{ij}}$ is the best configuration under which (i,j)-th entry is a feasible entry. Let the (i,j)-th entry of the final estimate $\hat{\boldsymbol{X}}$ be weighted average of the (i,j)-th entry of $\hat{\boldsymbol{X}}_k$,

$$\hat{\boldsymbol{X}}[i,j] = \frac{\sum_{k=1}^{d\vee d_{ij}} w_k \nu_{ijk} \hat{\boldsymbol{X}}_k[i,j]}{\sum_{k=1}^{d\vee d_{ij}} w_k \nu_{ijk}} \tag{11.10}$$

where w_k is the assigned weight to configuration C_k. One may simply use constant weights. A more precise approach is to use a set of weights that reflects the accuracy of each configuration such as $\mathrm{IC}(C_k)$.

The model average estimator take the weighted average of the recovered entries under the best d configurations for most of the missing entries. If an entry (i,j) is infeasible under all d configurations $C_1,\ldots,C_d$, the above estimator finds the best configuration (in terms of IC) under which (i,j)-th entry is a feasible entry, and fill the entry with the recovered entry under that configuration.

The benefit of model averaging is multi-fold. First it provides an effective approach to handle the infeasibility issue. The probability that there is an entry that is infeasible under all possible configurations is extremely small, for reasonably large (M,N), the number of factors of (P,Q). Hence the procedure is able to handle high missing rate. Second, model averaging provides potentially more robust and stable estimators, as demonstrated in many studies in statistics literature (Buckland et al., 1997; Raftery et al., 1997). Note that KPD is true for all possible configurations. With a finite rank r, they all provide an approximation to the signal matrix $\boldsymbol{X}$ in (11.3), with different qualities. Averaging over the best performing models potentially improves the quality of matrix recovery. Third, model averaging provides sharper resolution in the completed matrix, particularly in image reconstruction. Kronecker products induce a block structure in the resulting matrix hence often produce 'grainy' images. Averaging over several configurations reduces such effects. Fourth, although the

final predictive model after averaging is equivalent to a hybrid Kronecker product model

$$\boldsymbol{X} = \sum_{k=1}^{d} w_k \boldsymbol{X}_k,$$

where each $\boldsymbol{X}_k$ assumes a Kronecker product model (11.1) under configuration C_k. The model averaging approach bypasses the difficulty of jointly estimating such a model as well as determining the configurations in such a model. The effectiveness of the approach will be demonstrated in the empirical study.

In practice the matrix dimension P and Q may not have many factors, which limits the flexibility of the KPD approach as the candidate set $\mathcal{C}$ can be small. In this case it is possible to augment the observed matrix with additional missing rows and columns so that the new dimensions P^* and Q^* have more factors to expand the candidate set. is to make $P^* = 2^M$ and $Q^* = 2^N$. One can also use different P^* and Q^* as part of model averaging operation. With a good configuration determination procedure and effective model averaging, significant improvement in matrix completion tasks can be obtained.

11.4 Simulation

In this simulation experiment, we exam the performance of the configuration determination procedure under a rank-1 setting. Specifically, the data is generated in a random scheme, where $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ are realizations of Gaussian random matrices of size $2^m \times 2^n$ and size $2^{M-m} \times 2^{N-n}$, respectively, with i.i.d $N(0,1)$ entries. Let $\boldsymbol{X} = \lambda_0 \tilde{\boldsymbol{A}} \otimes \tilde{\boldsymbol{B}}$, where λ_0 is the parameter used to control the signal-to-noise ratio. After re-parametrization of above construction, we have the identifiable parameters as

$$\lambda = \lambda_0 \|\tilde{\boldsymbol{A}}\|_F \|\tilde{\boldsymbol{B}}\|_F, \quad \boldsymbol{A} = \frac{\tilde{\boldsymbol{A}}}{\|\tilde{\boldsymbol{A}}\|_F}, \quad \boldsymbol{B} = \frac{\tilde{\boldsymbol{B}}}{\|\tilde{\boldsymbol{B}}\|_F}.$$

The underlying complete matrix $\boldsymbol{Y}$ is generated according to $\boldsymbol{Y} = \boldsymbol{X} + \sigma \boldsymbol{E}$, where $\boldsymbol{E}$ contains i.i.d. standard normal entries. The observed set Ω is generated independently such that $P[(i,j) \in \Omega] = \tau$ for all (i,j).

In this simulation, we set $M = N = 9$ (hence the observed matrix is 512×512) and

IC_κ	τ	λ_0/σ					
		0.1	0.2	0.3	0.4	0.5	0.6
AIC	0.1	0	0	57	93	100	100
	0.2	0	0	98	100	100	100
	0.3	0	0	100	100	100	100
	0.6	100	100	100	100	100	100
	0.8	100	100	100	100	100	100
BIC	0.1	13	100	100	100	100	100
	0.2	40	100	100	100	100	100
	0.3	86	100	100	100	100	100
	0.6	100	100	100	100	100	100
	0.8	100	100	100	100	100	100

Table 11.1: Number of correct configuration selections over 100 repetitions for different λ_0/σ, τ and information criteria.

$(m_*, n_*) = (5, 4)$. A combination of six different λ_0/σ values and five different probabilities τ are considered. AIC ($\kappa = 2$) and BIC ($\kappa = (M+N)\ln 2$) are used. For each combination of λ_0/σ and τ, we repeat the simulation 100 times and record the number of repetitions that the true configuration (m_*, n_*) is selected by the minimization in (11.8). The result is reported in Table 11.1.

From the table, it is obvious (and intuitively true) that the information criteria perform better with smaller missing proportion (large τ) and larger signal-to-noise ratio. And BIC performs better than AIC in this setting for the more difficult cases when the τ is small and the signal-to-noise ratio λ_0/σ is small.

We estimate $\boldsymbol{X}$ with the configuration $(\hat{m}, \hat{n})$ selected by BIC and measure the error of recovered matrix $\hat{\boldsymbol{X}}$ by

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2 / \|\boldsymbol{X}\|_F^2,$$

which is the Frobenius norm of error matrix, normalized by the Frobenius norm of $\boldsymbol{X}$. The averaged error of $\hat{\boldsymbol{X}}$ over 100 repetitions are reported in Table 11.2.

It reveals that the estimation error decreases when more entries are observed and when the signal-to-noise ratio λ_0/σ increases. As a comparison, SVD matrix completion, whose rank is selected such that the number of parameters are similar to the one selected by BIC, is reported in Table 11.2 as well. We notice that, with (almost) the same number of parameters used, KPD matrix completion with configuration selected by BIC is uniformly

	τ	λ_0/σ					
		0.1	0.2	0.3	0.4	0.5	0.6
KPD	0.1	1.158	0.833	0.571	0.417	0.300	0.206
	0.2	1.032	0.688	0.518	0.384	0.272	0.181
	0.3	0.931	0.665	0.506	0.375	0.264	0.174
	0.6	0.841	0.650	0.498	0.367	0.257	0.167
	0.8	0.829	0.647	0.496	0.365	0.255	0.165
SVD	0.1	1.159	1.177	1.153	1.170	1.124	1.494
	0.2	1.042	1.042	1.042	1.043	1.046	1.050
	0.3	1.026	1.025	1.025	1.024	1.023	1.022
	0.6	1.010	1.010	1.008	1.004	0.999	0.995
	0.8	1.008	1.006	1.003	0.998	0.993	0.989

Table 11.2: Averaged error of $\hat{\boldsymbol{X}}$ over 100 repetitions.

better than its SVD counterpart when the true matrix $\boldsymbol{X}$ has a low Kronecker rank structure instead of low matrix rank structure. This simulation also reveals that it is not optimal to apply classical matrix completion on matrices with Kronecker product structure.

11.5 Example

In this section, we apply the matrix completion approach for Kronecker products to images based on the image of Lenna. The original color image is converted to grayscale as shown in Figure 11.2 (left image) such that the image can be represented by a 512×512 matrix $\boldsymbol{X}$ of real numbers between 0 and 1.

To simulate the process of noise corruption and missing data, we first add a noise matrix to the grayscale image such that $\boldsymbol{Y} = \boldsymbol{X} + \sigma \boldsymbol{E}$, where the entries of $\boldsymbol{E}$ are i.i.d. standard Gaussian errors. We set $\sigma = 0.1$ and the corrupted image $\boldsymbol{Y}$ is shown in the middle of Figure 11.2. We set the missing rate to 80% ($\tau = 0.2$) and generated the missing set Ω with i.i.d Bernoulli(τ).

Let $\boldsymbol{Y}^* = P_\Omega \boldsymbol{Y}$ be the observed matrix which is plotted in the right of Figure 11.2, where missing entries are filled with white. We follow the configuration determination procedure proposed in Section 11.2.3. Based on $P_\Omega \boldsymbol{Y}$, all information criteria MSE ($\kappa = 0$), AIC ($\kappa = 2$) and BIC ($\kappa = \ln(512 \times 512) = 18 \ln 2$) select the configuration $(\hat{p}_m, \hat{q}_n) = (32, 128)$, restricted to the feasible configuration set $\mathcal{C}_0$. It corresponds to decompose $\boldsymbol{X}$ to the Kronecker product of a 32×128 matrix and a 16×4 matrix. Recovered images using the configuration

Figure 11.2: (Left) Grayscale Lenna's image; (Middle) Lenna s image with noise; (Right) Noisy image with 20% observed entries.

and ranks 1 to 3 are shown in the left column in Figure 11.3. The face can be recognized from the rank-one recovered matrix and more details are added as the rank increases.

To compare the performance of matrix completion through KPD with the classical approach through low rank SVD, the observed matrix $P_\Omega \boldsymbol{Y}$ is fitted by the alternating minimization algorithm assuming a SVD structure, which is equivalent to applying the KPD matrix completion with the configuration $(512, 1)$. SVD matrix completion with ranks 4, 8 and 12 are fitted to match the numbers of parameters of ranks one to three of KPD matrix configuration $(32, 128)$. The recovered matrices from SVD matrix completion are shown in the right column in Figure 11.3. The superiority of the KPD approach is obvious from the images. Besides judging the recovered images by eyesight, we measure the error of recovered matrix by

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2 / \|\boldsymbol{X}\|_F^2,$$

where $\boldsymbol{X}$ is the image without noise and $\hat{\boldsymbol{X}}$ is the recovered matrix by matrix completion through either KPD or SVD. Table 11.3 reports the errors of fitted matrices from KPD approaches and the ones of SVD matrix completion with a similar number of parameters. It is clear that KPD matrix completion can recover the Lenna's image more accurately compared to the SVD matrix completion approach, with a similar number of parameters. The result is anticipated since KPD matrix completion has more flexibility in selecting the configurations, with SVD matrix completion being one of its special cases. The proposed configuration determination procedure is able to find a better configuration, which provides

KPD rank	1	2	3
Error	0.1558	0.1371	0.1707
SVD rank	4	8	12
Error	0.4045	0.2995	0.3236

Table 11.3: Error for Kronecker matrix completion and classical matrix completion with similar number of parameters.

better performance than one of its special cases. Table 11.3 also reveals that increasing the rank r does not always improve the performance as overfitting may occur. To test the performance of the model averaging approach, we consider several restricted candidate sets $\mathcal{C}_s$ defined in (11.9) with $s = 0.167, 0.028, 0.00079$, corresponding to $pq \wedge p^*q^* \geq 2^3, 2^4, 2^5$, respectively. We slightly abuse the notation and name them $\mathcal{C}_3$ to $\mathcal{C}_5$ for simplicity. We also consider both (Kronecker) rank 1 and 2 models. The configurations within the candidate sets are ordered based on BIC criterion and their corresponding recovered matrices are obtained. Using equal weights w_k, $\hat{\boldsymbol{X}}$ using different d, the number of configurations for model averaging, are obtained according to (11.10). The reconstruction errors of the completed matrices as a function of d are reported in Figure 11.4. The error rates using Kronecker rank $r = 2$ with restricted set $\mathcal{C}_3$ and $\mathcal{C}_4$ are worse than all the four lines presented in the figure and are not shown.

It can be seen from the figure that using rank $r = 2$ and restricted set $\mathcal{C}_5$ performs the best. The error of averaging the top six configurations is around 0.11, which is about 20% smaller than 0.1371 obtained by the rank-two single configuration KPD matrix completion in Table 11.3. The error rate stays roughly the same if more configurations are used in the averaging. The performance of rank-1 models continues to improve when the averaging includes more models. The configurations close to the corners actually provide better performance when only rank-1 models are used. Note that $\mathcal{C}_3$ includes the corner configurations 1×8 and 2×4 matrices. Using restricted set $\mathcal{C}_2$ (using 1×4 and 2×2) yields the same results as using $\mathcal{C}_3$, hence the extra configurations were not ranked in the top 10 models.

It is seen that rank-2 models are more sensitive to the corner cases. One reason is that under the configuration that contains completely missing blocks, a rank-two matrix completion is less robust compared with a rank-one model, resulting in severe overfitting.

Figure 11.3: (Left column) Recovered images using KPD (Right column) Recovered images using SVD.

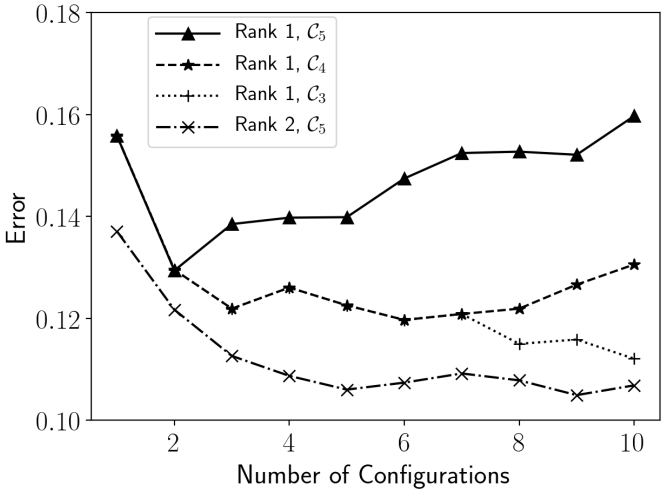


Figure 11.4: Reconstruction error of the averaged matrix against the number of configurations for four different scenarios.

When we include these configurations in model averaging, the results are poor.

The reconstructed images averaged over 10 rank-1 configurations under the restricted set $\mathcal{C}_3$ and over 9 rank-2 configurations under $\mathcal{C}_5$ are shown in Figure 11.5. Using the rank-1 average model, big pixels are observed in the reconstructed image but are less noisy than the ones in Figure 11.3. By using the rank-2 average model, more details are added resulting in a much smoother reconstructed image.

Figure 11.5: (Left) Average reconstructed image over 10 rank-one configurations. (Right) Average reconstructed image over 9 rank-two configurations.

Bibliography

Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., and Tandon, R. (2014). Learning sparsely used overcomplete dictionaries. In *Conference on Learning Theory*, pages 123–137.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105.

Ahn, S. C. and Horenstein, A. R. (2013). Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3):1203–1227.

Akaike, H. (1998). *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174.

Arora, S., Ge, R., Kannan, R., and Moitra, A. (2012). Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM.

Avitzour, D. (1995). Stochastic simulation bayesian approach to multitarget tracking. *IEE Proceedings-Radar, Sonar and Navigation*, 142(2):41–44.

Bai, J. (2003). Inferential theory for factor models of large dimensions. *Econometrica*, 71(1):135–171.

Baum, L. E. and Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Belliveau, J., Kennedy, D., McKinstry, R., Buchbinder, B., Weisskoff, R., Cohen, M., Vevea, J., Brady, T., and Rosen, B. (1991). Functional mapping of the human visual cortex by magnetic resonance imaging. *Science*, 254(5032):716–719.

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334.

Beskos, A., Papaspiliopoulos, O., Roberts, G. O., and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):333–382.

Binder, D. A. (1978). Bayesian cluster analysis. *Biometrika*, 65(1):31–38.

Bishop, M. and Thompson, E. A. (1986). Maximum likelihood alignment of dna sequences. *Journal of molecular biology*, 190(2):159–165.

Biswas, P., Lian, T.-C., Wang, T.-C., and Ye, Y. (2006). Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220.

Bound, J., Brown, C., and Mathiowetz, N. (2001). Measurement error in survey data. In *Handbook of Econometrics*, volume 5, pages 3705–3843. Elsevier.

Bruce, V. and Young, A. (1986). Understanding face recognition. *British journal of psychology*, 77(3):305–327.

Buckland, S. T., Burnham, K. P., and Augustin, N. H. (1997). Model selection: An integral part of inference. *Biometrics*, 53(2):603–618.

Cai, A., Tsay, R. S., and Chen, R. (2009). Variable selection in linear regression with many predictors. *Journal of Computational and Graphical Statistics*, 18(3):573–591.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982.

Cai, T. T., Zhang, A., et al. (2018). Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics. *The Annals of Statistics*, 46(1):60–89.

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

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

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

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). An improved particle for non-linear problems. *IEE Proceedings on Radar, Sonar, and Navigation*, 146:2–7.

Carroll, R., Ruppert, D., and Stefanski, L. (1995). Nonlinear measurement error models. *Monographs on Statistics and Applied Probability*, 63.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Chen, R., Wang, X., and Liu, J. S. (2000). Adaptive joint detection and decoding in flat-fading channels via mixture kalman filtering. *IEEE transactions on Information Theory*, 46(6):2079–2094.

Chen, R., Yang, D., and Zhang, C.-h. (2019a). Factor models for high-dimensional tensor time series. *arXiv preprint arXiv:1905.07530*.

Chen, X. and Xie, M. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, 24(4):1655–1684.

Chen, Y., Fan, J., Ma, C., and Yan, Y. (2019b). Inference and uncertainty quantification for noisy matrix completion. *Proceedings of the National Academy of Sciences*, 116(46):22931–22937.

Cheng, J. Q., Liu, R. Y., and Xie, M. (2017). Fusion learning. In Balakrishnan, N., Colton, T., Everitt, B., Piegorsch, W., Ruggeri, F., and Teugels, J. L., editors, *Wiley StatsRef: Statistics Reference Online*, pages 1–8. American Cancer Society.

Chiu, S.-T. (1991). Bandwidth selection for kernel density estimation. *The Annals of Statistics*, 19(4):1883–1905.

Cosse, A. and Demanet, L. (2017). Stable rank one matrix completion is solved by two rounds of semidefinite programming relaxation. *arXiv preprint arXiv:1801.00368*.

Crisan, D. and Doucet, A. (2000). Convergence of sequential monte carlo methods. *Signal Processing Group, Department of Engineering, University of Cambridge, Technical Report CUEDIF-INFENGrrR38*, 1.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Diaconis, P. and Freedman, D. (1986). On the consistency of bayes estimates. *The Annals of Statistics*, 14(1):1–26.

Doucet, A., Briers, M., and Sénécal, S. (2006). Efficient block sampling strategies for sequential monte carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Duarte, M. F. and Baraniuk, R. G. (2012). Kronecker compressive sensing. *IEEE Transactions on Image Processing*, 21(2):494–504.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Willey & Sons, New York.

Durham, G. B. and Gallant, A. R. (2002). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business & Economic Statistics*, 20(3):297–338.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56.

Fan, J., Liao, Y., and Mincheva, M. (2011). High dimensional covariance matrix estimation in approximate factor models. *Annals of statistics*, 39(6):3320.

Fan, J. and Truong, Y. K. (1993). Nonparametric regression with errors in variables. *The Annals of Statistics*, 21(4):1900–1925.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

Figueiredo, M. A. T. and Jain, A. K. (2000). Unsupervised learning of finite mixture models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24:381–396.

Fong, W., Godsill, S. J., Doucet, A., and West, M. (2002). Monte carlo smoothing with application to audio signal enhancement. *IEEE transactions on signal processing*, 50(2):438–449.

Foygel, R. and Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. In *Advances in neural information processing systems*, pages 604–612.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Fuller, W. A. (2009). *Measurement error models*, volume 305. John Wiley & Sons.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*, volume 20. Society for Industrial and Applied Mathematics.

Gao, X. and Carroll, R. J. (2017). Data integration with high dimensionality. *Biometrika*, 104(2):251–272.

Ghahramani, Z. and Jordan, M. I. (1996). Factorial hidden markov models. In *Advances in Neural Information Processing Systems*, pages 472–478.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target—monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.

Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational researcher*, 5(10):3–8.

Godsill, S., Doucet, A., and West, M. (2001). Maximum a posteriori sequence estimation using monte carlo particle filters. *Annals of the Institute of Statistical Mathematics*, 53(1):82–96.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–71.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., Airoldi, E. M., et al. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113.

Granville, V., Krivanek, M., and Rasson, J. . (1994). Simulated annealing: a proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):652–656.

Green, P. J. and Silverman, B. W. (1993). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press.

Guo, Q., Zhang, C., Zhang, Y., and Liu, H. (2015). An efficient SVD-based method for image denoising. *IEEE transactions on Circuits and Systems for Video Technology*, 26(5):868–880.

Hall, P., Park, B. U., and Samworth, R. J. (2008). Choice of neighbor order in nearest-neighbor classification. *The Annals of Statistics*, 36(5):2135–2152.

Hamburg, M. A. and Collins, F. S. (2010). The path to personalized medicine. *New England Journal of Medicine*, 363(4):301–304.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hodrick, R. J. and Prescott, E. C. (1997). Postwar us business cycles: an empirical investigation. *Journal of Money, credit, and Banking*, pages 1–16.

Hooke, R. and Jeeves, T. A. (1961). " direct search" solution of numerical and statistical problems. *J. ACM*, 8(2):212–229.

Hull, J. (2015). *Options, Futures, and Other Derivatives*. Pearson.

Insel, T. R. (2009). Translating scientific opportunity into public health impact: a strategic plan for research on mental illness. *Archives of general psychiatry*, 66(2):128–133.

Irie, K. and West, M. (2016). Bayesian emulation for optimization in multi-step portfolio decisions. *arXiv preprint arXiv:1607.01631*.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323.

Jain, P., Kar, P., et al. (2017). Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336.

Jain, P., Netrapalli, P., and Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM.

Jensen, C. S., Kjærulff, U., and Kong, A. (1995). Blocking gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies*, 42(6):647–666.

Joe, H. and Lee, Y. (2009). On weighting of bivariate margins in pairwise likelihood. *Journal of Multivariate Analysis*, 100(4):670–685.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python.

Juang, B.-H. (1984). On the hidden markov model and dynamic time warping for speech recognition - a unified view. *AT&T Bell Laboratories Technical Journal*, 63(7):1213–1243.

Juang, B. H. and Rabiner, L. R. (1991). Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kamm, J. and Nagy, J. G. (1998). Kronecker product and SVD approximations in image restoration. *Linear Algebra and its Applications*, 284(1):177 – 192. International Linear

Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing.

Kaye, P., Laflamme, R., Mosca, M., et al. (2007). *An introduction to quantum computing*. Oxford University Press.

Keshavan, R. H., Montanari, A., and Oh, S. (2010). Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078.

Kiefer, J., Wolfowitz, J., et al. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466.

Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with arch models. *The review of economic studies*, 65(3):361–393.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. (2009). \ell_1 trend filtering. *SIAM review*, 51(2):339–360.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25.

Koenig, L. L., Lucero, J. C., and Perlman, E. (2008). Speech production variability in fricatives of children and adults: Results of functional data analysis. *The Journal of the Acoustical Society of America*, 124(5):3158–3170.

Kolm, P. N. and Ritter, G. (2015). Multiperiod portfolio selection and bayesian dynamic models. *Risk*.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288.

Kyle, A. and Obizhaeva, A. (2011). Market microstructure invariants: Theory and implications of calibration. Available at SSRN: https://ssrn.com/abstract=1978932.

Land, A. H. and Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520.

Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338.

Le, C. M., Levina, E., Vershynin, R., et al. (2016). Optimization via low-rank approximation for community detection in networks. *The Annals of Statistics*, 44(1):373–400.

Liao, T. W. (2005). Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874.

Lin, M., Chen, R., and Liang, J. (2008a). Statistical geometry of lattice chain polymers with voids of defined shapes: Sampling with strong constraints. *The Journal of Chemical Physics*, 128:1–12.

Lin, M., Chen, R., and Liu, J. (2013). Lookahead strategies for Sequential Monte Carlo. *Statistical Science*, 28:69–94.

Lin, M., Chen, R., and Mykland, P. (2010). On generating monte carlo samples of continuous diffusion bridges. *Journal of the American Statistical Association*, 105(490):820–838.

Lin, M., Lu, H.-M., Chen, R., and Liang, J. (2008b). Generating properly weighted ensemble of conformations of proteins from sparse or indirect distance constraints. *The Journal Of Chemical Physics*, 129,094101:1–13.

Lin, M. T., Zhang, J. L., Cheng, Q., and Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association*, 100(472):1412–1421.

Lindsay, B. G. (1995). Mixture models: Theory, geometry and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics*, 5:i–163.

Liu, D., Liu, R., and Xie, M. (2014). Exact meta-analysis approach for discrete data and its application to 2×2 tables with rare events. *Journal of the American Statistical Association*, 109:1450–1465.

Liu, D., Liu, R. Y., and Xie, M. (2015). Multivariate meta-analysis of heterogeneous studies using only summary statistics: Efficiency and robustness. *Journal of the American Statistical Association*, 110(509):326–340.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the american statistical association*, 90(430):567–576.

Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044.

Liu, K. and Meng, X. L. (2016). There is individualized treatment. why not individualized inference? *Annual Review of Statistics and Its Application*, 3:79–111.

Lo, A. Y. (1984). On a class of bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, 12(1):351–357.

Maldjian, J. A., Laurienti, P. J., Kraft, R. A., and Burdette, J. H. (2003). An automated method for neuroanatomic and cytoarchitectonic atlas-based interrogation of fMRI data sets. *Neuroimage*, 19(3):1233–1239.

Marden, J. I. (1991). Sensitive and sturdy p-values. *The Annals of Statistics*, 19(2):918–934.

Markowitz, H. (1959). Portfolio selection, cowles foundation monograph no. 16. *John Wiley, New York*, 32:263–74.

Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Nakagawa, S. and Nakanishi, H. (1988). Speaker-independent english consonant and japanese word recognition by a stochastic dynamic time warping method. *IETE Journal of Research*, 34(1):87–95.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The computer journal*, 7(4):308–313.

Netrapalli, P., Niranjan, U., Sanghavi, S., Anandkumar, A., and Jain, P. (2014). Non-convex robust pca. In *Advances in Neural Information Processing Systems*, pages 1107–1115.

Ng, R. T. and Han, J. (1994). Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 144–155. Morgan Kaufmann Publishers Inc.

Ng, V., Engle, R. F., and Rothschild, M. (1992). A multi-dynamic-factor model for stock returns. *Journal of Econometrics*, 52(1-2):245–266.

Normand, S.-L. T. (1999). Meta-analysis: formulating, evaluating, combining, and reporting. *Statistics in Medicine*, 18(3):321–359.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198.

Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In *bmvc*, volume 1, page 6.

Pedersen, A. R. (1995). Consistency and asymptotic normality of an approximate maximum likelihood estimator for discretely observed diffusion processes. *Bernoulli*, pages 257–279.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

Qian, M. and Murphy, S. A. (2011). Performance guarantees for individualized treatment rules. *The Annals of Statistics*, 39(2):1180.

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191.

Robbins, H. (1956). An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 157–163. University of California Press.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Scharth, M. and Kohn, R. (2016). Particle efficient importance sampling. *Journal of Econometrics*, 190:133–147.

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464.

Schweder, T. and Hjort, N. L. (2016). *Confidence, Likelihood, Probability: Statistical Inference with Confidence Distributions*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Shen, J., Liu, R., and Xie, M. (2019). iFusion: Individualized fusion learning. *Journal of the American Statistical Association*. to appear, availabel at https://doi.org/10.1080/01621459.2019.1672557.

Singh, K., Xie, M., and Strawderman, W. E. (2005). Combining information from independent sources through confidence distributions. *The Annals of Statistics*, 33(1):159–183.

Stefanski, L. A. and Carroll, R. J. (1990). Deconvolving kernel density estimators. *Statistics*, 21(2):169–184.

Stratonovich, R. L. (1965). Conditional markov processes. In *Non-linear transformations of stochastic processes*, pages 427–453. Elsevier.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in Neural Information Processing Systems 17*, pages 1385–1392. MIT Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Turk, M. A. and Pentland, A. P. (1991). Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE.

Tzikas, D. G., Likas, A. C., and Galatsanos, N. P. (2008). The variational approximation for bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146.

Van der Laan, M. J. and Rose, S. (2011). *Targeted learning: causal inference for observational and experimental data.* Springer Science & Business Media.

van der Laan, M. J. and Rubin, D. (2006). Targeted maximum likelihood learning. *U.C. Berkeley Division of Biostatistics Working Paper Series.* Working Paper 213.

Van Loan, C. F. and Pitsianis, N. (1993). Approximation with kronecker products. In *Linear algebra for large scale and real-time applications*, pages 293–314. Springer.

Varin, C. and Vidoni, P. (2006). Pairwise likelihood inference for ordinal categorical time series. *Computational Statistics & Data Analysis*, 51:2365–2373.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing.*

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Wang, X., Chen, R., and Guo, D. (2002). Delayed-pilot sampling for mixture kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, 50(2):241–254.

Wansbeek, T. J. and Meijer, E. (2000). *Measurement error and latent variables in econometrics*, volume 37. North-Holland.

Wasserman, L. (2010). *All of Nonparametric Statistics*. Springer Publishing Company, Incorporated.

Wedin, P.-Å. (1972). Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111.

Werner, K., Jansson, M., and Stoica, P. (2008). On estimation of covariance matrices with kronecker product structure. *IEEE Transactions on Signal Processing*, 56(2):478–491.

West, M. and Harrison, J. (1998). Bayesian forecasting and dynamic models (2nd edn). *Journal of the Operational Research Society*, 49(2):179–179.

Whiteley, N. and Lee, A. (2014). Twisted particle filters. *Annals of Statistics*, 42:115–141.

Xie, M., Liu, R. Y., Damaraju, C. V., and Olson, W. H. (2013). Incorporating external information in analyses of clinical trials with binary outcomes. *The Annals of Applied Statistics*, 7(1):342–368.

Xie, M., Singh, K., and Strawderman, W. E. (2011). Confidence distributions and a unifying framework for meta-analysis. *Journal of the American Statistical Association*, 106(493):320–333.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

Yang, G., Liu, D., Wang, J., and Xie, M. (2016). Meta-analysis framework for exact inferences with application to the analysis of rare events. *Biometrics*, 72(4):1378–1386.

Yuan, M. and Zhang, C.-H. (2016). On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068.

Zhang, J., Dundas, J., Lin, M., Chen, R., Wang, W., and Liang, J. (2009). Prediction of geometrically feasible three dimensional structures of Pseudoknotted RNA through free energy. *RNA*, 15:2248–2263.

Zhang, J., Lin, M., Chen, R., Liang, J., and Liu, J. S. (2007). Monte carlo sampling of near-native structures of proteins with applications. *Proteins: Structure, Function, and Bioinformatics*, 66:61–68.

Zhang, J. L. and Liu, J. S. (2002). A new sequential importance sampling method and its application to the two-dimensional hydrophobic–hydrophilic model. *The Journal of Chemical Physics*, 117(7):3492–3498.

Zhao, Y., Zeng, D., Rush, A. J., and Kosorok, M. R. (2012). Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118.

Appendices

APPENDIX A

Theorem Proofs for PART II

Proof of Theorem 7.2

We prove the consistency first. Define

$$\begin{aligned}
\psi_k(\theta) &= \frac{\partial}{\partial\theta} M_k(\theta), \\
\Psi_K(\theta) &= \frac{\sum_{k=0}^{K} \mathcal{K}\left(\frac{\|\boldsymbol{z}_k - \boldsymbol{z}_0\|}{b}\right) \psi_k(\theta)}{\sum_{k=0}^{K} \mathcal{K}\left(\frac{\|\boldsymbol{z}_k - \boldsymbol{z}_0\|}{b}\right)}, \\
\Psi(\theta) &= \mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}_0} \psi_{\boldsymbol{x}}(\theta).
\end{aligned}$$

For any given θ, $\Psi_K(\theta)$ is a kernel smoothing estimator for $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}_0}[\psi_{\boldsymbol{x}}(\theta)] = \Psi(\theta)$. Hence $\Psi_K(\theta) \to \Psi(\theta)$ in probability for any given θ, provided $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}}[\psi_{\boldsymbol{x}}(\theta)]$ continuous at $\boldsymbol{z}_0$ (Wasserman, 2010). Due to the assumption that $M_k(\theta)$ is convex and second-order differentiable, $\psi_k(\theta)$ is a non-decreasing function for any $\boldsymbol{x}_k$. Therefore, both Ψ_K and Ψ are non-decreasing and continuous. By assumption, Θ_0 is the unique root of $\Psi(\theta)$. Let θ_K^* be such a point that $\Psi_K(\theta_K^*) = 0$. θ_K^* may not be unique and may not even exist for small K. For any $\epsilon > 0$, it is immediate that $\Psi(\Theta_0 - \epsilon) < 0 < \Psi(\Theta_0 + \epsilon)$ and by the pointwise convergence in probability of Ψ_K, we have

$$\begin{aligned}
P\left[|\Psi_K(\Theta_0 - \epsilon) - \Psi(\Theta_0 - \epsilon)| \leqslant \frac{1}{2}|\Psi(\Theta_0 - \epsilon)|\right] &\longrightarrow 1, \\
P\left[|\Psi_K(\Theta_0 + \epsilon) - \Psi(\Theta_0 + \epsilon)| \leqslant \frac{1}{2}|\Psi(\Theta_0 + \epsilon)|\right] &\longrightarrow 1.
\end{aligned}$$

Therefore,

$$P\Big[\,|\Psi_K(\Theta_0-\epsilon)-\Psi(\Theta_0-\epsilon)|\leqslant\frac{1}{2}\,|\Psi(\Theta_0-\epsilon)|\,,\\ |\Psi_K(\Theta_0+\epsilon)-\Psi(\Theta_0+\epsilon)|\leqslant\frac{1}{2}\,|\Psi(\Theta_0+\epsilon)|\Big]\longrightarrow 1.$$

The event in the probability implies that $\Psi_K(\Theta_0-\epsilon)<0<\Psi_K(\Theta_0+\epsilon)$, which further implies the existence of θ_K^* in $(\Theta_0-\epsilon,\Theta_0+\epsilon)$ by continuity of Ψ_K. Hence

$$\begin{aligned}&P\left[|\Psi_K(\Theta_0-\epsilon)-\Psi(\Theta_0-\epsilon)|\leqslant\frac{1}{2}\,|\Psi(\Theta_0-\epsilon)|\,,|\Psi_K(\Theta_0+\epsilon)-\Psi(\Theta_0+\epsilon)|\leqslant\frac{1}{2}\,|\Psi(\Theta_0+\epsilon)|\right]\\ \leqslant&P\left[\Psi_K(\Theta_0-\epsilon)<0<\Psi_K(\Theta_0+\epsilon)\right]\\ \leqslant&P\left[\Theta_0-\epsilon<\theta_K^*<\Theta_0+\epsilon\right].\end{aligned}$$

Since the first term converges to 1, the last term converges to 1 as well. Note that when $\tilde{\theta}_0^{(c)}$ exists, it equals θ_K^*. The consistency of $\tilde{\theta}_0^{(c)}$ is proved.

With $\tilde{\theta}_0^{(c)}\longrightarrow\theta_0$ in probability, it is reasonable to expand $\Psi_K(\tilde{\theta}_0^{(c)})$ at Θ_0.

$$\sum_{k=0}^{K}\mathcal{K}\left(\frac{\|\boldsymbol{z}_k-\boldsymbol{z}_0\|}{b}\right)\psi_k(\Theta_0)+(\tilde{\theta}^{(c)}-\Theta_0)\sum_{k=0}^{K}\mathcal{K}\left(\frac{\|\boldsymbol{z}_k-\boldsymbol{z}_0\|}{b}\right)\psi_k'(\Theta_0)\\ +\sum_{k=0}^{K}\mathcal{K}\left(\frac{\|\boldsymbol{z}_k-\boldsymbol{z}_0\|}{b}\right)O((\tilde{\theta}_0^{(c)}-\Theta_0)^2)=0.$$

Now we have

$$\tilde{\theta}_0^{(c)}-\Theta_0=-\frac{\frac{1}{K+1}\sum_{k=0}^{K}\mathcal{K}\left(\frac{\|\boldsymbol{z}_k-\boldsymbol{z}_0\|}{b}\right)\psi_k(\Theta_0)}{\frac{1}{K+1}\sum_{k=0}^{K}\mathcal{K}\left(\frac{\|\boldsymbol{z}_k-\boldsymbol{z}_0\|}{b}\right)\psi_k'(\Theta_0)+O(\hat{\theta}_0^{(c)}-\Theta_0)}.\tag{A.1}$$

Consider $K\to\infty$. On one hand, the numerator is a kernel smoothing estimator for $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}_0}[\psi_{\boldsymbol{x}}(\Theta_0)]=0$ up to a normalizing constant. On the other hand, the denominator is a similar kernel smoothing estimator for $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}_0}\psi_{\boldsymbol{x}}'(\Theta_0)$. By Slutsky's theorem, their ratio has a similar asymptotic distribution to the numerator kernel smoothing estimator up to a constant factor of $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{z}_0}\psi_{\boldsymbol{x}}'(\Theta_0)$. Therefore, $\tilde{\theta}_0^{(c)}$ has an asymptotic bias $O_p(b^2)$ and an asymptotic variance $O_p(1/Kb^d)$ (Wasserman, 2010). Hence, the optimal choice of band-

width in a bias-variance optimization scheme is $\hat{b} \asymp K^{-1/(d+4)}$ and the optimal MSE is of order $K^{-4/(d+4)}$.

Proof of Theorem 7.3

In this case, θ_0 is assumed to be fixed, and $\hat{\theta}_0^{(c)}$ is a standard kernel smoothing estimator for $\mathbb{E}_\pi[\theta_0|\boldsymbol{z}_0] = \Theta_0$. By following the asymptotic property of a standard kernel smoothing estimator, we have

$$\mathbb{E}[\hat{\theta}_0^{(c)}|\boldsymbol{z}_0] = \Theta_0 + O_p(b^2) \text{ and } Var[\hat{\theta}_0^{(c)}|\boldsymbol{z}_0] = O_p\left(\frac{1}{Kd^b}\right).$$

Therefore, we have

$$\begin{aligned}
&\mathbb{E}_{\theta_0}[\hat{\theta}_0^{(c)}] = \mathbb{E}_{\theta_0}[\Theta_0] + O_p(b^2), \\
&Var_{\theta_0}[\hat{\theta}_0^{(c)}] = Var_{\theta_0}[\mathbb{E}[\hat{\theta}_0^{(c)}|\boldsymbol{z}_0]] + \mathbb{E}_{\theta_0}[Var[\hat{\theta}_0^{(c)}|\boldsymbol{z}_0]] = Var_{\theta_0}[\Theta_0] + O_p\left(\frac{1}{Kd^b}\right).
\end{aligned}$$

Proof of Theorem 7.4

We first prove the following lemma, which would be used in the proof of Theorem 7.4.

Lemma A.1. *Suppose the random vector ξ has a pdf p_ξ and has zero mean, finite variance and finite higher moments such that*

$$\mathbb{E}\xi = 0, \quad Var(\xi) = \sigma^2\boldsymbol{\Sigma}, \quad \|\boldsymbol{\Sigma}\| = 1.$$

Then for any second-order partially differentiable function f, we have

$$\int f(\boldsymbol{x}+t)p_\xi(t)dt = f(\boldsymbol{x}) + \frac{1}{2}\sigma^2 \mathrm{tr}[\nabla^2 f(\boldsymbol{x})\boldsymbol{\Sigma}] + o(\sigma^2),$$

when $\sigma^2 \to 0$.

Proof. Let $\xi_1 = \xi/\sigma$, then $\mathbb{E}(\xi_1) = 0$ and $\mathrm{Var}(\xi_1) = \boldsymbol{\Sigma}$. Hence

$$\begin{aligned}
\int f(\boldsymbol{x}+t)p_\xi(t)dt &= \int f(\boldsymbol{x}+\sigma s)p_{\xi_1}(s)ds \\
&= \int \left[f(\boldsymbol{x}) + \sigma s^T[\nabla f(\boldsymbol{x})] + \frac{1}{2}\sigma^2 s^T[\nabla^2 f(x)]s + o(\sigma^2) \right] p_{\xi_1}(s)ds \\
&= f(\boldsymbol{x}) + \frac{1}{2}\sigma^2 \int s^T[\nabla^2 f(\boldsymbol{x})]s p_{\xi_1}(s)ds + o(\sigma^2) \\
&= f(\boldsymbol{x}) + \frac{1}{2}\sigma^2 \mathrm{tr}[\nabla^2 f(\boldsymbol{x})\boldsymbol{\Sigma}] + o(\sigma^2).
\end{aligned}$$

□

Now we prove Theorem 7.4. Let $\bar{\pi}()$ be the population distribution for $\boldsymbol{\eta}$. Since $\theta = g(\boldsymbol{\eta})$, we have

$$\begin{aligned}
\mathbb{E}_\pi[\theta_0|\boldsymbol{z}_0] &= \frac{\int g(\boldsymbol{\eta})p(\boldsymbol{z}_0|\boldsymbol{\eta})\bar{\pi}(\boldsymbol{\eta})d\boldsymbol{\eta}}{\int p(\boldsymbol{z}_0|\boldsymbol{\eta})\bar{\pi}(\boldsymbol{\eta})d\boldsymbol{\eta}} \\
&= \frac{(g\bar{\pi})(\boldsymbol{z}_0) + \frac{1}{2}\sigma_z^2 \mathrm{tr}[\nabla^2 (g\bar{\pi})(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z] + o(\sigma_z^2)}{\bar{\pi}(\boldsymbol{z}_0) + \frac{1}{2}\sigma_z^2 \mathrm{tr}[\nabla^2 \bar{\pi}(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z] + o(\sigma_z^2)} \\
&= \frac{(g\bar{\pi})(\boldsymbol{z}_0) + \frac{1}{2}\sigma_z^2 \mathrm{tr}[\nabla^2 (g\bar{\pi})(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z] + o(\sigma_z^2)}{\bar{\pi}(\boldsymbol{z}_0)} \left[1 - \frac{1}{2}\sigma_z^2 \frac{\mathrm{tr}[\nabla^2 \bar{\pi}(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z]}{\bar{\pi}(\boldsymbol{z}_0)} + o(\sigma_z^2) \right] \\
&= g(\boldsymbol{z}_0) + \frac{\sigma_z^2}{2\bar{\pi}(\boldsymbol{z}_0)} \left(\mathrm{tr}[\nabla^2 (g\bar{\pi})(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z] - g(\boldsymbol{z}_0)\mathrm{tr}[\nabla^2 \bar{\pi}(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z] \right) + o(\sigma_z^2) \\
&= g(\boldsymbol{z}_0) + \sigma_z^2 \left(\frac{\mathrm{tr}[\nabla^2 g(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z]}{2} + \frac{\mathrm{tr}[\nabla \bar{\pi}(\boldsymbol{z}_0)^T \boldsymbol{\Sigma}_z \nabla g(\boldsymbol{z}_0)]}{\bar{\pi}(\boldsymbol{z}_0)} \right) + o(\sigma_z^2).
\end{aligned}$$

Thus, the bias is

$$\begin{aligned}
&\mathbb{E}_{\theta_0}[\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0]] - g(\boldsymbol{\eta}_0) \\
=&\int \mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0]p(\boldsymbol{z}_0|\boldsymbol{\eta}_0)d\boldsymbol{z}_0 - g(\boldsymbol{\eta}_0) \\
=&\int \left(g(\boldsymbol{z}_0) + \sigma_z^2 \left(\frac{\mathrm{tr}[\nabla^2 g(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z]}{2} + \frac{\mathrm{tr}[\nabla \bar{\pi}(\boldsymbol{z}_0)^T \boldsymbol{\Sigma}_z \nabla g(\boldsymbol{z}_0)]}{\bar{\pi}(\boldsymbol{z}_0)} \right) + o(\sigma_z^2) \right) p(\boldsymbol{z}_0|\boldsymbol{\eta}_0)d\boldsymbol{z}_0 - g(\boldsymbol{\eta}_0) \\
=&g(\boldsymbol{\eta}_0) + \sigma_z^2 \left(\frac{\mathrm{tr}[\nabla^2 g(\boldsymbol{\eta}_0)\boldsymbol{\Sigma}_z]}{2} + \frac{\mathrm{tr}[\nabla \bar{\pi}(\boldsymbol{\eta}_0)^T \boldsymbol{\Sigma}_z \nabla g(\boldsymbol{\eta}_0)]}{\bar{\pi}(\boldsymbol{\eta}_0)} \right) + \frac{1}{2}\sigma_z^2 \mathrm{tr}[\nabla^2 g(\boldsymbol{\eta}_0)\boldsymbol{\Sigma}_z] + o(\sigma_z^2) - g(\boldsymbol{\eta}_0) \\
=&\sigma_z^2 \left(\mathrm{tr}[\nabla^2 g(\boldsymbol{\eta}_0)\boldsymbol{\Sigma}_z] + \frac{\mathrm{tr}[\nabla \bar{\pi}(\boldsymbol{\eta}_0)^T \boldsymbol{\Sigma}_z \nabla g(\boldsymbol{\eta}_0)]}{\bar{\pi}(\boldsymbol{\eta}_0)} \right) + o(\sigma_z^2)
\end{aligned}$$

$\asymp \sigma_z^2$.

On the other hand,

$$(\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0])^2 = g^2(\boldsymbol{z}_0) + \sigma_z^2 \left[g\mathrm{tr}[\nabla^2 g(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z] + \frac{2g\mathrm{tr}[\nabla\bar{\pi}(\boldsymbol{z}_0)^T\boldsymbol{\Sigma}_z\nabla g(\boldsymbol{z}_0)]}{\bar{\pi}(\boldsymbol{z}_0)} \right] + o(\sigma_z^2),$$

hence

$$\begin{aligned}
&\mathbb{E}_{\theta_0}\left[(\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0])^2\right] \\
=&\int (\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0])^2 p(\boldsymbol{z}_0|\boldsymbol{\eta}_0)d\boldsymbol{z}_0 \\
=&\int \left(g^2(\boldsymbol{z}_0) + \sigma_z^2 \left[g\mathrm{tr}[\nabla^2 g(\boldsymbol{z}_0)\boldsymbol{\Sigma}_z] + \frac{2g\mathrm{tr}[\nabla\bar{\pi}(\boldsymbol{z}_0)^T\boldsymbol{\Sigma}_z\nabla g(\boldsymbol{z}_0)]}{\bar{\pi}(\boldsymbol{z}_0)} \right] + o(\sigma_z^2) \right) p(\boldsymbol{z}_0|\boldsymbol{\eta}_0)d\boldsymbol{z}_0 \\
=&g^2(\boldsymbol{\eta}_0) + \sigma_z^2 \left[g\mathrm{tr}[\nabla^2 g(\boldsymbol{\eta}_0)\boldsymbol{\Sigma}_z] + \frac{2g\mathrm{tr}[\nabla\bar{\pi}(\boldsymbol{\eta}_0)^T\boldsymbol{\Sigma}_z\nabla g(\boldsymbol{\eta}_0)]}{\bar{\pi}(\boldsymbol{\eta}_0)} \right] + \frac{1}{2}\sigma_z^2\mathrm{tr}[\nabla^2(g^2)(\boldsymbol{\eta}_0)\boldsymbol{\Sigma}_z] + o(\sigma_z^2) \\
=&g^2(\boldsymbol{\eta}_0) + \sigma_z^2 \left[2g\mathrm{tr}[\nabla^2 g(\boldsymbol{\eta}_0)\boldsymbol{\Sigma}_z] + \frac{2g\mathrm{tr}[\nabla\bar{\pi}(\boldsymbol{\eta}_0)^T\boldsymbol{\Sigma}_z\nabla g(\boldsymbol{\eta}_0)]}{\bar{\pi}(\boldsymbol{\eta}_0)} + \mathrm{tr}[\nabla g(\boldsymbol{\eta}_0)^T\boldsymbol{\Sigma}_z\nabla g(\boldsymbol{\eta}_0)] \right] \\
&+ o(\sigma_z^2).
\end{aligned}$$

Therefore, the variance is

$$\begin{aligned}
\mathrm{Var}_{\theta_0}[\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0]] &= \mathbb{E}_{\theta_0}\left[(\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0])^2\right] - [\mathbb{E}_{\theta_0}[\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0]|\boldsymbol{\eta}_0]]^2 \\
&= \sigma_z^2\nabla g(\boldsymbol{\eta}_0)^T\boldsymbol{\Sigma}_z\nabla g(\boldsymbol{\eta}_0) + o(\sigma_z^2) \\
&\asymp \sigma_z^2.
\end{aligned}$$

Proof of Theorem 7.5

From Theorem 7.3, we have

$$\mathbb{E}_{\theta_0}[(\hat{\theta}_0^{(c)} - \theta_0)^2] = B_0^2 + 2B_0O_p(b^2) + O_p(b^4) + V_0 + O_p\left(\frac{1}{Kd^b}\right).$$

On the other hand,

$$\begin{aligned}
\mathbb{E}[B_0] &= \mathbb{E}[\mathbb{E}_{\theta_0}[\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0]] - g(\boldsymbol{\eta}_0)] = \mathbb{E}[g(\boldsymbol{\eta})] - \mathbb{E}[g(\boldsymbol{\eta}_0)] = 0, \\
\mathbb{E}[B_0^2 + V_0] &= Var[B_0] + \mathbb{E}[V_0] = Var[\mathbb{E}_{\theta_0}[\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0]] - \theta_0] + \mathbb{E}[Var_{\theta_0}[E_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0]]] \\
&= Var[\mathbb{E}_{\theta_0}[\mathbb{E}_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0] - \theta_0]] + \mathbb{E}[Var_{\theta_0}[E_\pi[g(\boldsymbol{\eta})|\boldsymbol{z}_0] - \theta_0]] \\
&= Var[\mathbb{E}_\pi[\theta_0|\boldsymbol{z}_0] - \theta_0].
\end{aligned}$$

Therefore,

$$\mathbb{E}[(\hat{\theta}_0^{(c)} - \theta_0)^2] = \mathbb{E}[\mathbb{E}_{\theta_0}[(\hat{\theta}_0^{(c)} - \theta_0)^2]] = Var[\mathbb{E}_\pi[\theta_0|\boldsymbol{z}_0] - \theta_0] + O_p(b^4) + O_p\left(\frac{1}{Kd^b}\right).$$

Proof of Theorem 7.6

The combined estimator can be written as

$$\hat{\theta}_0^{(c)} = \frac{\sum_{k=0}^K w(\hat{\theta}_k, \hat{\theta}_0)\hat{\theta}_k}{\sum_{k=0}^K w(\hat{\theta}_k, \hat{\theta}_0)} = \frac{\frac{1}{K+1}\sum_{k=0}^K w(\hat{\theta}_k, \hat{\theta}_0)\hat{\theta}_k}{\frac{1}{K+1}\sum_{k=0}^K w(\hat{\theta}_k, \hat{\theta}_0)}.$$

Let

$$q(\hat{\theta}) = \int p(\hat{\theta})\pi(\theta)d\theta.$$

By law of large number, when $K \to \infty$, the numerator is

$$\begin{aligned}
\frac{1}{K+1}\sum_{k=0}^K w(\hat{\theta}_k, \hat{\theta}_0)\hat{\theta}_k &\xrightarrow{P} \mathbb{E}[w(\hat{\theta}, \hat{\theta}_0)\hat{\theta}] \\
&= \int \left(\frac{1}{q(\hat{\theta})q(\hat{\theta}_0)}\int p(\hat{\theta}|\theta')p(\hat{\theta}_0|\theta')\pi(\theta')d\theta'\right)\hat{\theta}q(\hat{\theta})d\hat{\theta} \\
&= \frac{1}{q(\hat{\theta}_0)}\int \left(\int g(\hat{\theta})p(\hat{\theta}|\theta')d\hat{\theta}\right)p(\hat{\theta}_0|\theta')\pi(\theta')d\theta' \\
&= \frac{1}{q(\hat{\theta}_0)}\int \theta' p(\hat{\theta}_0|\theta')\pi(\theta')d\theta' \\
&= \int \theta'\pi(\theta'|\hat{\theta}_0)d\theta'.
\end{aligned}$$

Similarly, for the denominator, we have

$$\begin{aligned}
\frac{1}{K+1}\sum_{k=0}^{K} w(\hat{\theta}_k, \hat{\theta}_0) &\xrightarrow{P} \mathbb{E}[w(\hat{\theta}, \hat{\theta}_0)] \\
&= \int \left(\frac{1}{q(\hat{\theta}) q(\hat{\theta}_0)} \int p(\hat{\theta}|\theta') p(\hat{\theta}_0|\theta') \pi(\theta') d\theta' \right) q(\hat{\theta}) d\hat{\theta} \\
&= \frac{1}{q(\hat{\theta}_0)} \int \left(\int p(\hat{\theta}|\theta') d\hat{\theta} \right) p(\hat{\theta}_0|\theta') \pi(\theta') d\theta' \\
&= \frac{1}{q(\hat{\theta}_0)} \int p(\hat{\theta}_0|\theta') \pi(\theta') d\theta' \\
&= 1.
\end{aligned}$$

Hence, the combined estimator would converge in probability to the Bayes estimator with squared loss. On one hand, by central limit theorem, the numerator has asymptotic normality, provided finite second moment. On the other hand, the denominator converges to 1 in probability. By Slutsky's theorem, the ratio is also asymptotically normal with the same rate as central limit theorem. Therefore,

$$\sqrt{K}(\hat{\theta}_0^{(c)} - \mathbb{E}[\theta|\hat{\theta}_0]) = O_p(1).$$

Proof of Theorem 7.7

When $K \to \infty$, the target function in optimization is now

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K} w(\hat{\theta}_k, \hat{\theta}_0) f(\theta, \hat{\theta}_k) \\
&\xrightarrow{P} \int \frac{1}{q(\hat{\theta}) q(\hat{\theta}_0)} \left(\int p(\hat{\theta}|\theta') p(\hat{\theta}_0|\theta') \pi(\theta') d\theta' \right) f(\theta, \hat{\theta}) q(\hat{\theta}) d\hat{\theta} \\
&= \frac{1}{q(\hat{\theta}_0)} \int \left(\int f(\theta, \hat{\theta}) p(\hat{\theta}|\theta') d\hat{\theta} \right) p(\hat{\theta}_0|\theta') \pi(\theta') d\theta' \\
&= \frac{1}{q(\hat{\theta}_0)} \int L(\theta, \theta') p(\hat{\theta}_0|\theta') \pi(\theta') d\theta' + \frac{1}{q(\hat{\theta}_0)} \int C(\theta') p(\hat{\theta}_0|\theta') \pi(\theta') d\theta'.
\end{aligned}$$

The second component here is a constant with respect to θ. Given the assumptions on $M(\theta,\hat{\theta})$ and following the proof in Appendix A, we have

$$\arg\min_{\theta}\sum_{k=0}^{K} w(\hat{\theta}_k,\hat{\theta}_0)M(\theta,\hat{\theta}_k) \xrightarrow{P} \arg\min_{\theta}\int L(\theta,\theta')p(\hat{\theta}_0|\theta')\pi(\theta')d\theta' = \Theta_0.$$

Here, we simply denote the target estimator $\Theta_0(\boldsymbol{x}_0;L)$ as Θ_0. Let $M'_\theta(\theta,\hat{\theta}) = \dfrac{\partial M(\theta,\hat{\theta})}{\partial\theta}$, $M''_\theta(\theta,\hat{\theta}) = \dfrac{\partial^2 M(\theta,\hat{\theta})}{\partial\theta^2}$ and $\theta^*_K = \arg\min_\theta \sum_{k=0}^{K} w(\hat{\theta}_k,\hat{\theta}_0)M(\theta,\hat{\theta}_k)$. Then we have

$$\sum_{k=0}^{K} w(\hat{\theta}_k,\hat{\theta}_0)M'_\theta(\theta^*_K,\hat{\theta}_k) = 0.$$

Since θ^*_K converges to Θ_0, it's reasonable to expand the equation at Θ_0.

$$\sum_{k=0}^{K} w(\hat{\theta}_k,\hat{\theta}_0)M'_\theta(\Theta_0,\hat{\theta}_k) + (\theta^*_K-\Theta_0)\sum_{k=0}^{K} w(\hat{\theta}_k,\hat{\theta}_0)M''_\theta(\Theta_0,\hat{\theta}_k) + O_p((\theta^*_K-\Theta_0)^2) = 0.$$

Then

$$\theta^*_K - \Theta_0 = -\frac{\sum_{k=0}^{K} w(\hat{\theta}_k,\hat{\theta}_0)f'_\theta(\Theta_0,\hat{\theta}_k)}{\sum_{k=0}^{K} w(\hat{\theta}_k,\hat{\theta}_0)f''_\theta(\Theta_0,\hat{\theta}_k) + O_p(\theta^*_K-\Theta_0)}.$$

Given the numerator has a finite variance, by central limit theorem and Slutsky's theorem, it is immediate that

$$\sqrt{K}(\theta^*_K-\Theta_0) = O_p(1).$$

Proof of Theorem 7.8

Similar to the proof of Theorem 7.4, when $\sigma^2_\theta \to 0$, we have

$$\begin{aligned}
\Theta_0[\boldsymbol{x}_0;\ell_2] &= \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0] \\
&= \frac{\int \theta_0 p(\hat{\theta}_0 \mid \theta_0)\pi(\theta_0)d\theta_0}{\int \theta_0 p(\hat{\theta}_0 \mid \theta_0)\pi(\theta_0)d\theta_0} \\
&= \frac{\hat{\theta}_0\pi(\hat{\theta}_0) + \dfrac{1}{2}\sigma^2_\theta(\hat{\theta}_0\pi(\hat{\theta}_0))'' + o_p(\sigma^2_\theta)}{\pi(\hat{\theta}_0) + \dfrac{1}{2}\sigma^2_\theta(\pi(\hat{\theta}_0))'' + o_p(\sigma^2_\theta)}
\end{aligned}$$

$$= \hat{\theta}_0 + \frac{1}{2}\sigma_\theta^2 \left(\frac{(\hat{\theta}_0 \pi(\hat{\theta}_0))''}{\pi(\hat{\theta}_0)} - \frac{\hat{\theta}_0 (\pi(\hat{\theta}_0))''}{\pi(\hat{\theta}_0)} \right) + o_p(\sigma_\theta^2)$$
$$= \hat{\theta}_0 + \sigma_\theta^2 \frac{\pi'(\hat{\theta}_0)}{\pi(\hat{\theta}_0)} + o_p(\sigma_\theta^2).$$

Therefore, for any fixed θ_0

$$\begin{aligned}\mathbb{E}_{\theta_0}[\Theta_0[\boldsymbol{x}_0; \ell_2]] &= \int \left(\hat{\theta}_0 + \sigma_\theta^2 \frac{\pi'(\hat{\theta}_0)}{\pi(\hat{\theta}_0)} + o_p(\sigma_\theta^2) \right) p(\hat{\theta}_0 \mid \theta_0) d\hat{\theta}_0 \\ &= \theta_0 + \sigma_\theta^2 \frac{\pi'(\theta_0)}{\pi(\theta_0)} + \frac{1}{2}\sigma_\theta^2 \left(\theta_0 + \sigma_\theta^2 \frac{\pi'(\theta_0)}{\pi(\theta_0)} \right)'' + o_p(\sigma_\theta^2) \\ &= \theta_0 + \sigma_\theta^2 \frac{\pi'(\theta_0)}{\pi(\theta_0)} + o_p(\sigma_\theta^2),\end{aligned}$$

and similarly,

$$\mathbb{E}_{\theta_0}[\Theta_0[\boldsymbol{x}_0; \ell_2]^2] = \theta_0^2 + 2\sigma_\theta^2 \frac{\theta_0 \pi'(\theta_0)}{\pi(\theta_0)} + \sigma_\theta^2 + o_p(\sigma_\theta^2).$$

Hence, the bias is

$$B_0(\theta_0) = \mathbb{E}_{\theta_0}[\Theta_0[\boldsymbol{x}_0; \ell_2]] - \theta_0 = \sigma_\theta^2 \frac{\pi'(\theta_0)}{\pi(\theta_0)} + o_p(\sigma_\theta^2) \asymp \sigma_\theta^2,$$

and the variance is

$$V_0(\theta_0) = \mathbb{E}_{\theta_0}[\Theta_0[\boldsymbol{x}_0; \ell_2]^2] - \mathbb{E}_{\theta_0}[\Theta_0[\boldsymbol{x}_0; \ell_2]]^2 = \sigma_\theta^2 + o_p(\sigma_\theta^2) \asymp \sigma_\theta^2.$$

Proof of Theorem 7.9

The iGroup estimator is

$$\hat{\theta}_0^{(c)} = \frac{\frac{1}{K+1} \sum_{k=0}^{K} w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0) \hat{\theta}_k}{\frac{1}{K+1} \sum_{k=0}^{K} w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0)}.$$

When $K \to \infty$, the numerator converges to

$$\frac{1}{K+1} \sum_{k=0}^{K} w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0) \hat{\theta}_k$$

$$
\begin{aligned}
&\xrightarrow{P} \mathbb{E}[w(\hat{\theta}, \boldsymbol{z}; \hat{\theta}_0, \boldsymbol{z}_0)\hat{\theta}] \\
&= \iint \mathcal{K}\left(\frac{\|\boldsymbol{z} - \boldsymbol{z}_0\|}{b}\right) \frac{\int p(\hat{\theta}|\theta)p(\hat{\theta}_0|\theta)p(\theta|\boldsymbol{z}_0)d\theta}{p(\hat{\theta}|\boldsymbol{z})p(\hat{\theta}_0|\boldsymbol{z}_0)} \hat{\theta} p(\hat{\theta}, \boldsymbol{z}) d\hat{\theta} d\boldsymbol{z} \\
&= \frac{1}{p(\hat{\theta}_0|\boldsymbol{z}_0)} \left(\iint p(\hat{\theta}|\theta)p(\hat{\theta}_0|\theta)p(\theta|\boldsymbol{z}_0)\hat{\theta} d\theta d\hat{\theta}\right) \left(\int \mathcal{K}\left(\frac{\|\boldsymbol{z} - \boldsymbol{z}_0\|}{b}\right) p(\boldsymbol{z}) d\boldsymbol{z}\right) \\
&\xrightarrow{P} \frac{p(\boldsymbol{z}_0)}{p(\hat{\theta}_0|\boldsymbol{z}_0)} \iint p(\hat{\theta}|\theta)p(\hat{\theta}_0|\theta)p(\theta|\boldsymbol{z}_0) g(\hat{\theta}) d\theta d\hat{\theta} \\
&= \frac{p(\boldsymbol{z}_0)}{p(\hat{\theta}_0|\boldsymbol{z}_0)} \int \left(\int p(\hat{\theta}|\theta) g(\hat{\theta}) d\hat{\theta}\right) p(\hat{\theta}_0|\theta)p(\theta|\boldsymbol{z}_0) d\theta \\
&= \frac{p(\boldsymbol{z}_0)}{p(\hat{\theta}_0|\boldsymbol{z}_0)} \int \theta p(\hat{\theta}_0|\theta)p(\theta|\boldsymbol{z}_0) d\theta \\
&= p(\boldsymbol{z}_0) \int \theta p(\theta|\hat{\theta}_0, \boldsymbol{z}_0) d\theta.
\end{aligned}
$$

Similarly for the denominator, we have

$$
\frac{1}{K+1} \sum_{k=0}^{K} w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0) \xrightarrow{P} \mathbb{E}[w(\hat{\theta}, \boldsymbol{z}; \hat{\theta}_0, \boldsymbol{z}_0)] \xrightarrow{P} p(\boldsymbol{z}_0).
$$

Therefore, the ratio converges to the target estimator $\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0|\hat{\theta}_0, \boldsymbol{z}_0]$. Moreover, by central limit theorem, given bandwidth b, the numerator has an error of order $1/\sqrt{K}$:

$$
\frac{1}{K+1} \sum_{k=0}^{K} w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0)\hat{\theta}_k - \mathbb{E}[w(\hat{\theta}, \boldsymbol{z}; \hat{\theta}_0, \boldsymbol{z}_0)\hat{\theta}] = O_p\left(K^{-1/2}\right).
$$

It brings a zero bias bias and a $O_p(1/K)$ variance. Now consider the kernel smoothing part, which yields a bias of order b^2 and a variance of order $1/(Kb^d)$. Therefore, the overall bias is of order b^2 and the overall variance is of order $O_p(K^{-1}) + O_p(1/(Kb^d)) = O_p(1/(Kb^d))$. Both the bias and variance is of the same order as in a d-dimensional kernel smoothing estimator. Hence, the optimal choice of the bandwidth is $\hat{b} \asymp K^{1/(d+4)}$, under which the optimal mean squared error is $O_p(K^{-4/d+4})$.

Proof of Theorem 7.10

When $K \to \infty$, the combined objective function is

$$
\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K} w(\hat{\theta}_k, \boldsymbol{z}_k; \hat{\theta}_0, \boldsymbol{z}_0) M(\theta, \hat{\theta}_k) \\
&\xrightarrow{P} \iint \mathcal{K}\left(\frac{\|\boldsymbol{z}-\boldsymbol{z}_0\|}{b}\right) \frac{\int p(\hat{\theta}|\theta')p(\hat{\theta}_0|\theta')p(\theta'|\boldsymbol{z}_0)d\theta'}{p(\hat{\theta}|\boldsymbol{z})p(\hat{\theta}_0|\boldsymbol{z}_0)} M(\theta,\hat{\theta})p(\hat{\theta},\boldsymbol{z})d\hat{\theta}d\boldsymbol{z} \\
&= \frac{1}{p(\hat{\theta}_0|\boldsymbol{z}_0)}\left(\iint p(\hat{\theta}|\theta')p(\hat{\theta}_0|\theta')p(\theta'|\boldsymbol{z}_0)M(\theta,\hat{\theta})d\theta' d\hat{\theta}\right)\left(\int \mathcal{K}\left(\frac{\|\boldsymbol{z}-\boldsymbol{z}_0\|}{b}\right)p(\boldsymbol{z})d\boldsymbol{z}\right) \\
&\xrightarrow{P} \frac{p(\boldsymbol{z}_0)}{p(\hat{\theta}_0|\boldsymbol{z}_0)} \iint p(\hat{\theta}|\theta')p(\hat{\theta}_0|\theta')p(\theta'|\boldsymbol{z}_0)M(\theta,\hat{\theta})d\theta' d\hat{\theta} \\
&= \frac{p(\boldsymbol{z}_0)}{p(\hat{\theta}_0|\boldsymbol{z}_0)} \int \left(\int p(\hat{\theta}|\theta')M(\theta,\hat{\theta})d\hat{\theta}\right) p(\hat{\theta}_0|\theta')p(\theta'|\boldsymbol{z}_0)d\theta' \\
&= \frac{p(\boldsymbol{z}_0)}{p(\hat{\theta}_0|\boldsymbol{z}_0)} \int (L(\theta,\theta')+C(\theta'))p(\hat{\theta}_0|\theta')p(\theta'|\boldsymbol{z}_0)d\theta' \\
&= p(\boldsymbol{z}_0)\int L(\theta,\theta')p(\theta'|\hat{\theta}_0,\boldsymbol{z}_0)d\theta' + p(\boldsymbol{z}_0)\int C(\theta')p(\theta'|\hat{\theta}_0,\boldsymbol{z}_0)d\theta'.
\end{aligned}
$$

The second term here is a constant with respect to θ. Given the convex and second-order differentiable condition of $M(\theta,\hat{\theta})$, following the proof in Appendix A, the iGroup estimator converges to the target estimator $\Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; L)$ in probability. Given the consistency, one can expand the term at Θ_0 as in Appendix A (proof of Theorem 7.7) except that the weight is replaced by the full weight $w(\hat{\theta}, \boldsymbol{z}; \hat{\theta}_0, \boldsymbol{z}_0)$. By following the same argument in Appendix A, the numerator has an asymptotic mean squared error of order $K^{-4/(d+4)}$ when the bandwidth is chosen to be optimal $\hat{b} \asymp K^{-1/(d+4)}$. Provided the denominator converges in probability to its expectation by law of large number, we have $\tilde{\theta}_0^{(c)}$ has a mean squared error of order $K^{-4/(d+4)}$.

Proof of Theorem 7.11

For fixed σ_z^2, the result follows immediately from the proof of Theorem 7.8 except that $\Theta_0[\hat{\theta}_0; \ell 2] = \int \theta_0 p(\hat{\theta}_0 \mid \theta_0)\pi(\theta_0)d\theta_0 / \int p(\hat{\theta}_0 \mid \theta_0)\pi(\theta_0)d\theta_0$ is replaced by $\Theta_0[\hat{\theta}_0, \boldsymbol{z}_0; \ell 2] = \int \theta_0 p(\hat{\theta}_0 \mid \theta_0)p(\boldsymbol{z}_0 \mid \theta_0)\pi(\theta_0)d\theta_0 / \int p(\hat{\theta}_0 \mid \theta_0)p(\boldsymbol{z}_0 \mid \theta_0)\pi(\theta_0)d\theta_0$. For fixed σ_θ^2, the result follows from the same proof in Theorem 7.4.

Proof of Proposition 7.1

Consider the problem based on both information sets $\mathcal{D}_x$ and $\mathcal{D}_z$ and use the notation of the target estimator $\Theta_0 = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0]$. Notice that

$$\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \theta_0 = (\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \Theta_0) + (\Theta_0 - \theta_0).$$

Given any fixed $(\boldsymbol{x}_0, \boldsymbol{z}_0)$, the first term $\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \Theta_0$ depends on other individuals' observations $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_K)$, while the second term $\Theta_0 - \theta_0$ depends on the true parameter θ_0, which is treated as random. Therefore, these two terms are independent conditioned on $(\boldsymbol{x}_0, \boldsymbol{z}_0)$, and we have

$$\begin{aligned}\mathbb{E}[(\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \theta_0)^2 \mid \boldsymbol{x}_0, \boldsymbol{z}_0] =& \mathbb{E}[(\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \Theta_0)^2 \mid \boldsymbol{x}_0, \boldsymbol{z}_0] + \mathbb{E}[(\Theta_0 - \theta_0)^2 \mid \boldsymbol{x}_0, \boldsymbol{z}_0] \\ &+ 2\mathbb{E}[\delta_0(\mathcal{D}_x, \mathcal{D}_z) - \Theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0]\mathbb{E}[\Theta_0 - \theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0].\end{aligned}$$

The last term $\mathbb{E}[\Theta_0 - \theta_0 \mid \boldsymbol{x}_0, \boldsymbol{z}_0]$ is zero. By taking expectation over $\boldsymbol{x}_0$ and $\boldsymbol{z}_0$, the decomposition is proved. Similar procedure for information set $\mathcal{D}_x$ or $\mathcal{D}_z$.

Proof of Proposition 7.2

Consider the problem based on both information sets $\mathcal{D}_x$ and $\mathcal{D}_z$. We expand the loss function at $\hat{\theta} = \Theta_0$ such that

$$L(\delta_0, \theta_0) = L(\Theta_0, \theta_0) + (\delta_0 - \Theta_0)L'(\Theta_0, \theta_0) + \frac{1}{2}(\delta_0 - \Theta_0)^2 L''(\Theta_0, \theta_0) + o((\delta_0 - \Theta_0)^2). \quad \text{(A.2)}$$

Notice that

$$\mathbb{E}_\pi[(\delta_0 - \Theta_0)L'(\Theta_0, \theta_0) \mid \boldsymbol{x}_0, \boldsymbol{z}_0] = \mathbb{E}_\pi[(\delta_0 - \Theta_0) \mid \boldsymbol{x}_0, \boldsymbol{z}_0]\mathbb{E}_\pi[L'(\Theta_0, \theta_0) \mid \boldsymbol{x}_0, \boldsymbol{z}_0] = 0.$$

The first equality is because for fixed $\boldsymbol{x}_0$ and $\boldsymbol{z}_0$, $\delta_0 - \Theta_0$ depends on other individuals' observations and $L'(\Theta_0, \theta_0)$ depends on the value of θ_0. The second equality is because Θ_0 is the minimizer of $\mathbb{E}_\pi[L(\Theta_0, \theta_0) \mid \boldsymbol{x}_0, \boldsymbol{z}_0]$. Hence, by taking expectation of Equation (A.2),

we have the desired decomposition. Similar procedure for information set $\mathcal{D}_x$ or $\mathcal{D}_z$.

Proof of Proposition 7.3

We first calculate the mean squared error of $\hat{\theta}_0^{(c)}$ conditioned on $(\hat{\theta}_0, \boldsymbol{z}_0)$. Notice that

$$\hat{\theta}_0^{(c)} - \theta_0 = (\hat{\theta}_0^{(c)} - \Theta_0) + (\Theta_0 - \theta_0).$$

Given any fixed $(\hat{\theta}_0, \boldsymbol{z}_0)$, the first term $\hat{\theta}_0^{(c)} - \Theta_0$ is a function of other individuals' observations $(\hat{\theta}_1, \ldots, \hat{\theta}_K, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_K)$, while the second term $\Theta_0 - \theta_0$ is a function of the true parameter θ_0, which is treated as random. Therefore, these two terms are independent conditioned on $(\hat{\theta}_0, \boldsymbol{z}_0)$, and we have

$$\begin{aligned}\mathbb{E}[(\hat{\theta}_0^{(c)} - \theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] = \mathbb{E}[(\hat{\theta}_0^{(c)} - \Theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] + \mathbb{E}[(\Theta_0 - \theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] \\ + 2\mathbb{E}[\hat{\theta}_0^{(c)} - \Theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\mathbb{E}[\Theta_0 - \theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0].\end{aligned}$$

Furthermore, since Θ_0 is a function of $\hat{\theta}_0$ and $\boldsymbol{z}_0$, we have

$$\begin{aligned}\mathbb{E}[(\Theta_0 - \theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] &= \Theta_0^2 - 2\Theta_0\mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0] + \mathbb{E}[\theta_0^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] \\ &= (\Theta_0 - \mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0])^2 + \mathbb{E}[\theta_0^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] - \Big(\mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\Big)^2 \\ &= (\Theta_0 - \mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0])^2 + \mathbb{E}[(\mathbb{E}[\theta \mid \hat{\theta}_0, \boldsymbol{z}_0] - \theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0].\end{aligned}$$

Hence, the conditional mean squared error of $\hat{\theta}_0^{(c)}$ becomes

$$\begin{aligned}\mathbb{E}[(\hat{\theta}_0^{(c)} - \theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] =& \mathbb{E}[(\hat{\theta}_0^{(c)} - \Theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] + (\Theta_0 - \mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0])^2 \\ &+ \mathbb{E}[(\mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0] - \theta_0)^2 \mid \hat{\theta}_0, \boldsymbol{z}_0] \\ &+ 2\mathbb{E}[\hat{\theta}_0^{(c)} - \Theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\mathbb{E}[\Theta_0 - \theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0].\end{aligned}$$

By taking the expectation for $\hat{\theta}_0$ and $\boldsymbol{z}_0$ on both sides, we have

$$\mathbb{E}[(\hat{\theta}_0^{(c)} - \theta_0)^2] = \mathbb{E}[(\hat{\theta}_0^{(c)} - \Theta_0)^2] + \mathbb{E}(\Theta_0 - \mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0])^2 + \mathbb{E}[(\mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0] - \theta_0)^2]$$

$$
\begin{aligned}
&+2\mathbb{E}\left\{\mathbb{E}[\hat{\theta}_0^{(c)}-\Theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\mathbb{E}_{\hat{\theta}_0,\boldsymbol{z}_0}[\Theta_0-\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\right\} \\
=&R_{np}(\hat{\theta}_0^{(c)})+R_{inf}(\hat{\theta}_0^{(c)})+R_0+2\mathbb{E}\left\{\mathbb{E}[\hat{\theta}_0^{(c)}-\Theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\mathbb{E}[\Theta_0-\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\right\}.
\end{aligned} \tag{A.3}
$$

The only thing left is to show the last term is 0. When $\Theta_0 = \Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]$ as in Case 3 in Section 7.2.4, it is straightforward that

$$
\mathbb{E}[\Theta_0-\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0] = \Theta_0 - \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0] = 0.
$$

When $\Theta_0 = \Theta_0(\boldsymbol{x}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0]$ as in Case 2 in Section 7.2.3, neither $\hat{\theta}_0^{(c)}$ nor Θ_0 depends on $\boldsymbol{z}$, and we can prove it by taking expectation over $\boldsymbol{z}_0$ first as follows

$$
\begin{aligned}
\mathbb{E}\left\{\mathbb{E}[\hat{\theta}_0^{(c)}-\Theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\mathbb{E}[\Theta_0-\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\right\} &= \mathbb{E}\left\{\mathbb{E}[\hat{\theta}_0^{(c)}-\Theta_0 \mid \hat{\theta}_0]\mathbb{E}[\Theta_0-\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0]\right\} \\
&= \mathbb{E}\left\{\mathbb{E}[\hat{\theta}_0^{(c)}-\Theta_0 \mid \hat{\theta}_0]\mathbb{E}\left(\mathbb{E}[\Theta_0-\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0] \mid \hat{\theta}_0\right)\right\} \\
&= \mathbb{E}\left\{\mathbb{E}[\hat{\theta}_0^{(c)}-\Theta_0 \mid \hat{\theta}_0]\mathbb{E}\left(\Theta_0-\mathbb{E}[\theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0] \mid \hat{\theta}_0\right)\right\} \\
&= \mathbb{E}\left\{\mathbb{E}[\hat{\theta}_0^{(c)}-\Theta_0 \mid \hat{\theta}_0]\left(\Theta_0-\mathbb{E}_\pi[\theta_0 \mid \hat{\theta}_0]\right)\right\} \\
&= 0.
\end{aligned}
$$

Similarly, when $\Theta_0 = \Theta_0(\boldsymbol{z}_0; \ell_2) = \mathbb{E}_\pi[\theta_0 \mid \boldsymbol{z}_0]$ as in Case 1 in Section 7.2.2, it can be shown by taking expectation over $\hat{\theta}_0$ first. Therefore, for all cases we considered, the last term in (A.3) equals 0, and we have

$$
R(\hat{\theta}_0^{(c)}) = R_{np}(\hat{\theta}_0^{(c)}) + R_{inf}(\hat{\theta}_0^{(c)}) + R_0.
$$

Proof of Proposition 7.4

Since $\tilde{\theta}_0^{(c)} \to \Theta_0(\hat{\theta}_0, \boldsymbol{z}_0)$ for all $\hat{\theta}_0$ and $\boldsymbol{z}_0$, the loss function can be expanded at Θ_0 as follows

$$
L(\tilde{\theta}_0^{(c)}, \theta_0) = L(\Theta_0, \theta_0) + L'(\Theta_0, \theta_0)(\tilde{\theta}_0^{(c)} - \Theta_0) + \frac{1}{2}L''(\Theta_0, \theta_0)(\tilde{\theta}_0^{(c)} - \Theta_0)^2 + o_p((\tilde{\theta}_0^{(c)} - \Theta_0)^2).
$$

By taking expectation on both sides, we have

$$\begin{aligned}\mathbb{E}[L(\tilde{\theta}_0^{(c)},\theta_0)] &= \mathbb{E}[L(\Theta_0,\theta_0)] + \frac{1}{2}\mathbb{E}[L''(\Theta_0,\theta_0)(\tilde{\theta}_0^{(c)}-\Theta_0)^2] \\ &\quad + o(\mathbb{E}[(\tilde{\theta}_0^{(c)}-\Theta_0)^2]) + \mathbb{E}[L'(\Theta_0,\theta_0)(\tilde{\theta}_0^{(c)}-\Theta_0)] \\ &= (\tilde{R}_0 + \tilde{R}_{inf}(\tilde{\theta}_0^{(c)})) + \tilde{R}_{np}(\tilde{\theta}_0^{(c)}) + o(\mathbb{E}[(\tilde{\theta}_0^{(c)}-\Theta_0)^2]) + \mathbb{E}[L'(\Theta_0,\theta_0)(\tilde{\theta}_0^{(c)}-\Theta_0)].\end{aligned} \tag{A.4}$$

It only needs to show the last term is 0. When in Case 3, $\Theta_0 = \Theta_0(\boldsymbol{x}_0, \boldsymbol{z}_0; L)$, and $L'(\Theta_0,\theta_0)$ and $(\tilde{\theta}_0^{(c)} - \Theta_0)$ are independent conditioned on $(\hat{\theta}_0, \boldsymbol{z}_0)$. Therefore,

$$\mathbb{E}[L'(\Theta_0,\theta_0)(\tilde{\theta}_0^{(c)}-\Theta_0) \mid \hat{\theta}_0, \boldsymbol{z}_0] = \mathbb{E}[L'(\Theta_0,\theta_0) \mid \hat{\theta}_0, \boldsymbol{z}_0]\mathbb{E}[\tilde{\theta}_0^{(c)} - \Theta_0 \mid \hat{\theta}_0, \boldsymbol{z}_0].$$

The first term $\mathbb{E}[L'(\Theta_0,\theta_0) \mid \hat{\theta}_0, \boldsymbol{z}_0]$ equals 0 because $\Theta_0 = \arg\min_\theta \mathbb{E}_\pi[L(\theta,\theta_0) \mid \hat{\theta}_0, \boldsymbol{z}_0]$. Similarly, in Case 1 and Case 2, the conditional expectations of $L'(\Theta_0,\theta_0)$ conditioned on $\boldsymbol{z}_0$ and $\hat{\theta}_0$ respectively are 0. Hence, the last term in (A.4) is always 0.

Proof of Proposition 7.5

Noticing that $\hat{\theta}_{(-k)}^{(c)} - \theta_k$ and $\hat{\theta}_k^{(c)} - \theta_k$ are independent with each other, we have

$$\begin{aligned}\mathbb{E}(\hat{\theta}_{(-k)}^{(c)} - \hat{\theta}_k)^2 &= \mathbb{E}(\hat{\theta}_{(-k)}^{(c)} - \theta_k)^2 + \mathbb{E}(\hat{\theta}_k - \theta_k)^2 + 0 \\ &= \mathbb{E}\left(\hat{\theta}_k^{(c)} - \theta_k + \frac{w(k;k)}{\sum_{l\neq k} w(l;k)}(\hat{\theta}_k^{(c)} - \hat{\theta}_k)\right)^2 + \mathbb{E}(\hat{\theta}_k - \theta_k)^2 \\ &= \mathbb{E}\left(\hat{\theta}_k^{(c)} - \theta_k\right)^2 + \mathbb{E}(\hat{\theta}_k - \theta_k)^2 + O\left(\frac{1}{K}\right).\end{aligned}$$

Therefore, the expectation of cross validation error is

$$\begin{aligned}\mathbb{E}(CV_{\Omega_0}(b)) &= \frac{1}{|\Omega_0|}\sum_{k\in\Omega_0}\mathbb{E}(\hat{\theta}_{(-k)}^{(c)} - \hat{\theta}_k)^2 \\ &= \frac{1}{|\Omega_0|}\sum_{k\in\Omega_0}\left[\mathbb{E}\left(\hat{\theta}_k^{(c)} - \theta_k\right)^2 + \mathbb{E}(\hat{\theta}_k - \theta_k)^2 + O\left(\frac{1}{K}\right)\right] \\ &= R_K(b) + \mathbb{E}_{\Omega_0}(\hat{\theta} - \theta)^2 + O\left(\frac{1}{K}\right),\end{aligned}$$

where the second term is averaging over all individuals in Ω_0 and hence a constant term with respect to bandwidth.

APPENDIX B

Theorem Proofs for PART III

Proof of Theorem 9.1 and Corollary 9.1

Without loss of generality, we assume $\sigma = 1$. Noticing that

$$\hat{\lambda} = \|\mathcal{R}_{m_0,n_0}[\boldsymbol{Y}]\|_S = \|\lambda \text{vec}(\boldsymbol{A})\text{vec}(\boldsymbol{B})' + \sigma 2^{-(M+N)/2}\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]\|_S,$$

by triangular inequality, we have

$$\left|\hat{\lambda} - \|\lambda \text{vec}(\boldsymbol{A})\text{vec}(\boldsymbol{B})'\|_S\right| \leqslant \sigma 2^{-(M+N)/2}\|\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]\|_S,$$

where $\|\lambda \text{vec}(\boldsymbol{A})\text{vec}(\boldsymbol{B})'\|_S = \lambda$. The following bound for $\|\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]\|_S$ can be obtained using the concentration inequality from Vershynin (2010),

$$P(\|\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]\|_S \geqslant 2^{(m_0+n_0)/2} + 2^{(M+N-m_0-n_0)/2} + t) \leqslant e^{-t^2/2}.$$

Therefore, $\|\mathcal{R}_{m_0,n_0}[\boldsymbol{E}]\|_S = s_0 + O_p(1)$ and

$$|\hat{\lambda} - \lambda| \leqslant 2^{-(M+N)/2}(s_0 + O_p(1)) = r_0 + O_p(2^{-(M+N)/2}),$$

which yields $\hat{\lambda} - \lambda = O_p(r_0)$.

The bounds for $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ corresponds to the error bounds in estimating the left and right singular vectors of $\mathcal{R}_{m_0,n_0}[\boldsymbol{Y}]$, which is a direct consequence of the analysis in Wedin (1972)

by observing that

$$\|\hat{\boldsymbol{A}} - \boldsymbol{A}\|_F^2 = \|\mathrm{vec}(\hat{\boldsymbol{A}}) - \mathrm{vec}(\boldsymbol{A})\|_2^2 = 2\sin^2\Theta(\mathrm{vec}(\hat{\boldsymbol{A}}), \mathrm{vec}(\boldsymbol{A})).$$

A sharper bound is provided in Cai et al. (2018).

Since above analysis holds for any fixed value of λ, Corollary 9.1 follows immediately.

Proof of Theorem 9.2

We first show and prove several technical lemmas.

Lemma B.1. *Suppose $a_n > 0, a_n \to 0$ and $x_n = O_p(1)$ is a sequence of continuous random variables with density functions p_n satisfying*

(i) $\mathbb{E}|x_n| \leqslant C$ *for some constant C for every n,*

(ii) $1 + a_n x_n > 0$ *almost surely,*

(iii) $a_n^{-2} \sup_{x \leqslant -1/(2a_n)} p_n(x) \to 0$,

then we have

$$\mathbb{E}\ln(1 + a_n x_n) = O(a_n).$$

Proof. Let $p_n(x_n)$ be the density function of x_n. For the positive part, we have

$$E_+ = \int_0^{+\infty} \ln(1 + a_n t) p_n(t) dt \leqslant \int_0^{+\infty} a_n t p_n(t) dt \leqslant a_n \mathbb{E}|x_n| \leqslant C a_n.$$

For the negative part, we have

$$\begin{aligned}
E_- &= \int_{-1/a_n}^{0} \ln(1 + a_n t) p_n(t) dt \\
&= \int_{-1/a_n}^{-1/(2a_n)} \ln(1 + a_n t) p_n(t) dt + \int_{-1/(2a_n)}^{0} \ln(1 + a_n t) p_n(t) dt \\
&\geqslant \left[\sup_{t \leqslant -1/(2a_n)} p_n(t) \right] \int_{-1/a_n}^{-1/(2a_n)} \ln(1 + a_n t) dt + \int_{-1/(2a_n)}^{0} 2 a_n t p_n(t) dt \\
&\geqslant -\frac{1 + \ln 2}{2a_n} \sup_{t < -1/(2a_n)} p_n(t) + 2a_n \int_{-\infty}^{0} t p_n(t) dt
\end{aligned}$$

$$\geqslant o(a_n) - 2Ca_n.$$

Hence,

$$\mathbb{E}\ln(1 + a_n x_n) = E_+ + E_- = O(a_n).$$

□

The conditions in Lemma B.1 are easy to verify in the subsequent proofs. Condition (ii) ensures the logarithm is well-defined on the whole support. Condition (i) is satisfied when x_n converges in mean to a random variable x with finite expectation. Condition (iii) is controlling the left tails of the densities, and is easily fulfilled if they are exponential.

Lemma B.2. *Let $\boldsymbol{X}$ be an arbitrary $P \times Q$ real matrix with $P \leqslant Q$ and $\boldsymbol{E}$ be a $P \times Q$ matrix with IID standard Gaussian entries. Then we have*

$$\mathbb{E}\|\boldsymbol{X} + \boldsymbol{E}\|_S^2 \leqslant \|\boldsymbol{X}\|_S^2 + (\sqrt{P} + \sqrt{Q})^2 + 4\|\boldsymbol{X}\|_S\sqrt{P} + \sqrt{2\pi}(\sqrt{P} + \sqrt{Q}) + 2 =: U^2.$$

Furthermore, the departure from the expectation is sub-Gaussian such that for any positive t, we have

$$P[\|\boldsymbol{X} + \boldsymbol{E}\|_S \geqslant U + t] \leqslant e^{-t^2/2}.$$

Proof. Without loss of generality, we assume $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2]$, where $\boldsymbol{X}_1 \in \mathbb{R}^{P \times P}$ is a diagonal matrix and $\boldsymbol{X}_2 \in \mathbb{R}^{P \times (Q-P)}$ is zero. Such a form of $\boldsymbol{X}$ can always be achieved by multiplying $\boldsymbol{X}$ and $\boldsymbol{E}$ from left and right by orthogonal matrices, without changing the distribution of $\boldsymbol{E}$. Similarly, we partition $\boldsymbol{E}$ into $[\boldsymbol{E}_1, \boldsymbol{E}_2]$ with $\boldsymbol{E}_1 \in \mathbb{R}^{P \times P}$ and $\boldsymbol{E}_2 \in \mathbb{R}^{P \times (Q-P)}$. Then

$$\begin{aligned}
\|\boldsymbol{X} + \boldsymbol{E}\|_S^2 &= \sup_{u \in \mathbb{R}^P, \|u\| = 1} u'(\boldsymbol{X} + \boldsymbol{E})(\boldsymbol{X} + \boldsymbol{E})'u \\
&= \sup_{u \in \mathbb{R}^P, \|u\| = 1} u'\boldsymbol{X}\boldsymbol{X}'u + u'\boldsymbol{E}\boldsymbol{E}u + 2u'\boldsymbol{X}\boldsymbol{E}'u \\
&\leqslant \|\boldsymbol{X}\|_S^2 + \|\boldsymbol{E}\|_S^2 + 2\|\boldsymbol{X}\|_S\|\boldsymbol{E}_1\|_S
\end{aligned}$$

According to Vershynin (2010), we have $\mathbb{E}\|\boldsymbol{E}_1\|_S \leqslant 2\sqrt{P}$ and

$$P[\|\boldsymbol{E}\|_S \geqslant \sqrt{P}+\sqrt{Q}+t] \leqslant e^{-t^2/2}.$$

Therefore,

$$\mathbb{E}\|\boldsymbol{E}\|_S^2 = \int_{t=0}^{\infty} P[\|\boldsymbol{E}\|_S > t]2tdt \leqslant (\sqrt{P}+\sqrt{Q})^2 + \sqrt{2\pi}(\sqrt{P}+\sqrt{Q}) + 2.$$

Hence, we have

$$\mathbb{E}\|\boldsymbol{X}+\boldsymbol{E}\|_S^2 \leqslant \|\boldsymbol{X}\|_S^2 + (\sqrt{P}+\sqrt{Q})^2 + 4\|\boldsymbol{X}\|_S\sqrt{P} + \sqrt{2\pi}(\sqrt{P}+\sqrt{Q}) + 2 =: U^2.$$

Since for any fixed $\boldsymbol{X}$, $\|\boldsymbol{X}+\boldsymbol{E}\|_S$ is a function of $\boldsymbol{E}$ with Lipschitz norm 1, by concentration inequality, for any positive t, we have

$$P[\|\boldsymbol{X}+\boldsymbol{E}\|_S \geqslant U+t] \leqslant e^{-t^2/2}.$$

□

We rewrite the information criterion as

$$\mathrm{IC}_\kappa(m,n) = D\left[\ln\|\boldsymbol{Y}-\hat{\boldsymbol{Y}}^{(m,n)}\|_F^2 + \kappa r_{m,n}^2 - 2\kappa D^{-1/2}\right],$$

where $D = 2^{M+N}$ and $r_{m,n} = 2^{-(m+n)/2} + 2^{-(m^\dagger+n^\dagger)/2}$. The constant term $2\kappa D^{-1/2}$ is irrelevant to the configuration (m,n) and is therefore ignored in subsequent proofs. Without loss of generality, we define the following expected information criterion

$$\mathrm{EIC}_\kappa(m,n) = D\left[\mathbb{E}\ln\|\boldsymbol{Y}-\hat{\boldsymbol{Y}}^{(m,n)}\|_F^2 + \kappa r_{m,n}^2\right]$$

for simplicity. The difference in expected information criterion between wrong configurations and the true configuration is of central interest, so we define

$$\Delta\mathrm{EIC}_\kappa(m,n) = \mathrm{EIC}_\kappa(m,n) - \mathrm{EIC}_\kappa(m_0,n_0)$$

Under the true configuration (m_0, n_0), we have

$$\mathbb{E}\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}^{(m,n)}\|_F^2 \leqslant \mathbb{E}\|\boldsymbol{Y} - \lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2 = \sigma^2 D^{-1} \mathbb{E}\|\boldsymbol{E}\|_F^2 = \sigma^2.$$

Therefore, we have

$$\mathrm{EIC}_\kappa(m_0, n_0) \leqslant D\left[\ln \mathbb{E}\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}^{(m,n)}\|_F^2 + \kappa r_0^2\right] \leqslant D\left[\ln \sigma^2 + \kappa r_0^2\right], \tag{B.1}$$

where $r_0 = r_{m_0,n_0}$.

Define

$$\hat{\lambda}^{(m,n)} := \|\mathcal{R}_{m,n}[\boldsymbol{Y}]\|_S = \|\lambda \mathcal{R}_{m,n}[\boldsymbol{A} \otimes \boldsymbol{B}] + \sigma D^{-1/2} \mathcal{R}_{m,n}[\boldsymbol{E}]\|_S. \tag{B.2}$$

To calculate the information criterion for wrong configurations, we use the following equality

$$\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}^{(m,n)}\|_F^2 = \|\boldsymbol{Y}\|_F^2 - \left[\hat{\lambda}^{(m,n)}\right]^2.$$

Notice that

$$\|\boldsymbol{Y}\|_F^2 = \|\lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2 + \sigma^2 D^{-1} \|\boldsymbol{E}\|_F^2 + 2\lambda\sigma D^{-1/2} \mathrm{tr}[(\boldsymbol{A} \otimes \boldsymbol{B})\boldsymbol{E}'],$$

where $\|\lambda \boldsymbol{A} \otimes \boldsymbol{B}\|_F^2 = \lambda^2$, $\sigma^2 D^{-1} \|\boldsymbol{E}\|_F^2 = \sigma^2(1 + O_p(D^{-1/2}))$ and $\mathrm{tr}[(\boldsymbol{A} \otimes \boldsymbol{B})\boldsymbol{E}']$ follows a standard normal distribution. We have

$$\|\boldsymbol{Y}\|_F^2 = \lambda^2 + \sigma^2 + R_1, \tag{B.3}$$

where

$$R_1 = O_p\left((\sigma^2 + \lambda\sigma) D^{-1/2}\right).$$

For wrong configurations $(m, n) \in \mathcal{W}$, without loss of generality, we assume $m + n \leqslant (M + N)/2$. According to Lemma B.2, we have the upper bound for (B.2):

$$\begin{aligned}
[\hat{\lambda}^{(m,n)}]^2 &\leqslant \lambda^2 \phi^2 + \sigma^2 r_{m,n}^2 + 4\lambda\phi\sigma 2^{(m+n)/2} D^{-1/2} + O_p((\lambda\sigma + \sigma^2) D^{-1/2}) \\
&\leqslant \lambda^2 \phi^2 + \sigma^2 r_{m,n}^2 + 4\lambda\sigma D^{-1/4} + O_p((\lambda\sigma + \sigma^2) D^{-1/2}).
\end{aligned} \tag{B.4}$$

Hence,

$$\|\boldsymbol{Y}-\hat{\boldsymbol{Y}}^{(m,n)}\|_S^2 \geqslant \lambda^2(1-\phi^2)+\sigma^2(1-r_{m,n}^2)-4\lambda\sigma D^{-1/4}+O_p((\lambda\sigma+\sigma^2)D^{-1/2}).$$

The last two terms are minor terms by Assumption 9.3. Therefore,

$$\mathrm{EIC}_\kappa(m,n) \geqslant D\left[\ln(\lambda^2\psi^2+\sigma^2(1-r_{m,n}^2))-O\left(\frac{\lambda\sigma D^{-1/4}}{\sigma^2+\lambda^2\psi^2}\right)+\kappa r_{m,n}^2\right]. \tag{B.5}$$

Here Lemma B.1 is applied since the stochastic term in (B.4) has an exponential tail bound. Notice that $\mathrm{EIC}_\kappa(m,n)$ in (B.5) is either a monotone increasing function or a uni-modal function of $r_{m,n}^2$ on $[1/2, 4D^{1/2}]$. Therefore, the minimum of the right hand side of (B.5) is obtained on the boundary. When $r_{m,n}^2 = 1/2$, (B.5) becomes

$$\mathrm{EIC}_\kappa(m,n) \geqslant D\left[\ln(\lambda^2\psi^2+\sigma^2/2)-O\left(\frac{\lambda\sigma D^{-1/4}}{\sigma^2+\lambda^2\psi^2}\right)+\kappa/2\right]. \tag{B.6}$$

When $r_{m,n}^2 = 4D^{-1/2}$, (B.5) becomes

$$\mathrm{EIC}_\kappa(m,n) \geqslant D\left[\ln(\lambda^2\psi^2+\sigma^2)-O\left(\frac{\lambda\sigma D^{-1/4}}{\sigma^2+\lambda^2\psi^2}\right)\right]. \tag{B.7}$$

In conclusion, for any wrong configuration $(m,n)\in\mathcal{W}$, we have

$$\Delta\mathrm{EIC}_\kappa(m,n) \geqslant D\left[\alpha-O\left(\frac{\lambda\sigma D^{-1/4}}{\sigma^2+\lambda^2\psi^2}\right)-\kappa r_0^2,\right] \tag{B.8}$$

where

$$\alpha = \left[\ln\left(1+\frac{\lambda^2\psi^2}{\sigma^2}\right)\right] \wedge \left[\ln\left(\frac{1}{2}+\frac{\lambda^2\psi^2}{\sigma^2}\right)+\frac{\kappa}{2}\right].$$

When $\kappa \geqslant 2\ln 2$, α takes the first value in the preceding equation. The assumptions imposed in Theorem 9.2 ensure the leading term α in (B.8) dominates other terms so that the minimum of ΔEIC over the wrong configurations is strictly positive.

We now address Remark 6. It turns out possible to use only the MSE to select the configuration, which corresponds to $\kappa = 0$. It requires a stronger signal-to-noise ratio $\lambda^2\psi^2/\sigma^2 > 1/2$ so that the leading term α in (B.8) is positive, and hence Theorem 9.2

continues to hold.

Note that the upper bound used in (B.4) is quite conservative, because the maximums of ϕ and $2^{(m+n)/2}$ over $\mathcal{W}$ are taken separately. It leads to a simple form of Assumption 9.3, which is actually not as optimal as possible. If we define $\phi^{(m,n)} = \|\mathcal{R}_{m,n}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S$, then the condition (9.12) in Assumption 9.3 can be relaxed to

$$\lim_{M+N\to\infty}\inf_{(m,n)\in\mathcal{W}}(2^{(m+n)/2}+2^{(m^\dagger+n^\dagger)/2})\cdot\frac{\lambda}{\sigma}\cdot\frac{1-[\phi^{(m,n)}]^2}{\phi^{(m,n)}}=\infty.$$

However, in the main text we choose to introduce the concept of representation gap and present a simple version of Assumption 9.3.

Proof of Theorem 9.3

We begin with the tail bounds for $\|\boldsymbol{E}\|_F^2$. According to the tail bounds for χ^2 random variable given in Laurent and Massart (2000), it holds that for any $t>0$,

$$P\left[D^{-1}\|\boldsymbol{E}\|_F^2 > 1+\sqrt{2}D^{-1/2}t+D^{-1}t^2\right]\leqslant e^{-t^2/2}, \tag{B.9}$$

$$P\left[D^{-1}\|\boldsymbol{E}\|_F^2 < 1-\sqrt{2}D^{-1/2}t\right]\leqslant e^{-t^2/2}, \tag{B.10}$$

where $D=2^{M+N}$. Therefore, at the true configuration (m_0,n_0), we have

$$\begin{aligned}&P\left[\|\boldsymbol{Y}-\hat{\boldsymbol{Y}}^{(m_0,n_0)}\|_F^2>\sigma^2+\sqrt{2}\sigma^2D^{-1/2}t+\sigma^2D^{-1}t^2\right]\\ \leqslant&P\left[\|\sigma D^{-1/2}\boldsymbol{E}\|_F^2>\sigma^2+\sqrt{2}\sigma^2D^{-1/2}t+\sigma^2D^{-1}t^2\right]\\ \leqslant&e^{-t^2/2}.\end{aligned} \tag{B.11}$$

Noticing that

$$\|\boldsymbol{Y}\|_F^2=\lambda^2+\sigma^2D^{-1}\|\boldsymbol{E}\|_F^2+2\lambda\sigma D^{-1/2}Z,$$

where $Z=\mathrm{tr}[(\boldsymbol{A}\otimes\boldsymbol{B})\boldsymbol{E}']$ is a standard Gaussian random variable, by (B.10) we have

$$P\left[\|\boldsymbol{Y}\|_F^2<\lambda^2+\sigma^2-(\sqrt{2}\sigma^2+2\lambda\sigma)D^{-1/2}t\right]\leqslant 2e^{-t^2/2}. \tag{B.12}$$

Now we consider the tail bound for $\hat{\lambda}^{(m,n)}$ of wrong configurations. According to Lemma B.2, we have the tail bound for $\hat{\lambda}^{(m,n)}$ as

$$P[\hat{\lambda}^{(m,n)} \geqslant U + \sigma D^{-1/2} t] \leqslant e^{-t^2/2}, \tag{B.13}$$

where

$$U^2 = \lambda^2\phi^2 + \sigma^2 r_{m,n}^2 + 4\lambda\phi\sigma 2^{(m+n)/2} D^{-1/2} + \sqrt{2\pi}\sigma^2 r_{m,n} D^{-1/2} + 2\sigma^2 D^{-1} < (\lambda+\sigma)^2.$$

Let $\alpha = \ln(1 + (\lambda/\sigma)^2\psi^2)$ be the positive gap constant. We have

$$\begin{aligned} & P\left[\mathrm{IC}_\kappa(m_0,n_0) > \mathrm{EIC}_\kappa(m_0,n_0) + D\alpha/3\right] \\ =& P\left[\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}^{(m_0,n_0)}\|_F^2 > \sigma^2 e^{\alpha/3}\right] \\ \leqslant& \exp\left(-c_1^2 D/2\right), \end{aligned} \tag{B.14}$$

where

$$c_1^2 = e^{\alpha/3} - 1.$$

For any $(m,n) \in \mathcal{W}$, it holds that

$$\begin{aligned} & P\left[\mathrm{IC}_\kappa(m,n) < \mathrm{EIC}_\kappa(m_0,n_0) + D\alpha/3\right] \\ \leqslant& P\left[\mathrm{IC}_\kappa(m,n) < \mathrm{EIC}_\kappa(m,n) - D\alpha/3\right] \\ \leqslant& P\left[\|\boldsymbol{Y}\|_F^2 - \hat{\lambda}^2 < \lambda^2 + \sigma^2 - \lambda^2\phi^2 - 2h\right] \\ \leqslant& P\left[\|\boldsymbol{Y}\|_F^2 < \lambda^2 + \sigma^2 - h\right] + P\left[\hat{\lambda}^2 > U^2 + h\right] \\ \leqslant& 2\exp\left(-c_2^2 D/2\right) + \exp\left(-c_3^2 D/2\right) \end{aligned} \tag{B.15}$$

where we use (B.12) and (B.13) to obtain (B.15),

$$h = \frac{1}{2}\left(1 - e^{-\alpha/3}\right)(\lambda^2(1-\phi^2) + \sigma^2), \quad c_2 = \frac{h}{\sqrt{2\sigma^2 + 2\lambda\sigma}},$$

and c_3 is the solution of

$$\sigma^2 c_3^2 + 2(\lambda+\sigma)\sigma c_3 = h.$$

We conclude that

$$\begin{aligned}
&P\left[\mathrm{IC}_\kappa(m_0,n_0) \geqslant \min_{(m,n)\in\mathcal{W}} \mathrm{IC}_\kappa(m,n)\right] \\
\leqslant &\sum_{(m,n)\in\mathcal{W}} P[\mathrm{IC}_\kappa(m_0,n_0) \geqslant \mathrm{IC}_\kappa(m,n)] \\
\leqslant &\sum_{(m,n)\in\mathcal{W}} \Big(P[\mathrm{IC}_\kappa(m_0,n_0) \geqslant \mathrm{EIC}_\kappa(m_0,n_0) + D\alpha/3] \\
&\qquad + P[\mathrm{IC}_\kappa(m,n) \leqslant \mathrm{EIC}_\kappa(m_0,n_0) + D\alpha/3] \Big) \\
\leqslant &4(M+1)(N+1)\exp\left[-c^2 D/2\right] \to 0, \qquad \text{(B.16)}
\end{aligned}$$

where $c = \min\{c_1, c_2, c_3\}$. By calculating the orders of c_1, c_2, c_3, it holds that

$$c^2 \geqslant O\left((e^{\alpha/3}-1) \wedge \left(\frac{e^\alpha - e^{2\alpha/3}}{1+\lambda/\sigma} \right)^2 \right).$$

Specifically, if $\alpha \to 0$ (or equivalently, $(\lambda/\sigma)^2\psi^2 \to 0$), we have

$$c^2 \geqslant O\left(\frac{\lambda^2}{\sigma^2}\psi^2 \wedge \frac{(\lambda^2/\sigma^2)^2}{(1+\lambda/\sigma)^2}\psi^4 \right)$$

The right hand side is much greater than $\ln(MN)$, under Assumptions 9.1 and 9.3.

Proof of Theorem 9.4

The proof is very similar to the proofs of Theorem 9.2 and Theorem 9.3, so we only point out the major steps, but omit the details. Condition (9.15) implies that $\lambda^2 = \lambda_0^2(1+o_p(1))$ and $\psi^2 = \psi_0^2(1+o_p(1))$. The proof of Theorem 9.2 follows immediately by replacing λ^2 and ψ^2 with the deterministic values λ_0^2 and ψ_0^2, except that an $o_p(\lambda_0^2+\psi_0^2)$ term is added to (B.2). Since the additional stochastic term is negligible and has finite expectation, Theorem 9.2 continues to hold.

The consistency follows same lines as those of Theorem 9.3 except that the deviations $\lambda^2-\lambda_0^2$ and $\psi^2-\psi_0^2$ should be incorporated into (B.16). Specifically, Assumption 9.4 implies that for any small constant δ, with probability larger than $1-o(1/(MN))$, we have $\lambda^2 \geqslant \lambda_0^2(1-\delta)$ and $\psi^2 \geqslant \psi_0^2(1-\delta)$. Proof of Theorem 9.3 follows immediately by replacing λ^2 and ψ^2 with $\lambda_0^2(1-\delta)$ and $\psi_0^2(1-\delta)$. The following probability of exceptions should be added to (B.16).

$$(M+1)(N+1)\left[P[\lambda^2<\lambda_0^2(1-\delta)]+P[\psi^2<\psi_0^2(1-\delta)]\right]=o(1),$$

which does not affect consistency but may reduce the convergence rate.

Proof of Lemma 9.1 and Corollary 9.2

Consider the complete Kronecker product decomposition of $\boldsymbol{A}$ with respect to the configuration $(m\wedge m', n\wedge n', (m-m')_+, (n-n')_+)$:

$$\boldsymbol{A}=\sum_{i=1}^{I}\mu_i \boldsymbol{C}_i\otimes\boldsymbol{D}_i, \tag{B.17}$$

where $I=2^{m\wedge m'+n\wedge n'}\wedge 2^{(m-m')_+ +(n-n')_+}$, $\mu_1\geqslant\mu_2\geqslant\cdots\geqslant\mu_I$ are the coefficients in decreasing order. $\boldsymbol{C}_i$ and $\boldsymbol{D}_i$ satisfy

$$\langle \boldsymbol{C}_i,\boldsymbol{C}_j\rangle=\langle \boldsymbol{D}_i,\boldsymbol{D}_j\rangle=\delta_{i,j}, \tag{B.18}$$

where $\delta_{i,j}$ is the Kronecker delta function such that $\delta_{i,j}=1$ if and only if $i=j$ and $\delta_{i,j}=0$ otherwise, and $\langle \boldsymbol{A},\boldsymbol{B}\rangle := \mathrm{tr}[\boldsymbol{A}'\boldsymbol{B}]$ is the trace inner product. Notice that the decomposition in (B.17) corresponds to the singular value decomposition for $\mathcal{R}_{m\wedge m',n\wedge n'}[\boldsymbol{A}]$. Therefore, the singular values $\mu_1,\ldots,\mu_I$ are uniquely identifiable and the components $\boldsymbol{C}_i$, $\boldsymbol{D}_i$ are identifiable if the singular values are distinct. In particular,

$$\mu_1=\|\mathcal{R}_{m\wedge m',n\wedge n'}[\boldsymbol{A}]\|_S.$$

Similarly, the KPD of $\boldsymbol{B}$ with the configuration $((m'-m)_+,(n'-n)_+,M-m\vee m',N-n\vee n')$ is given by

$$\boldsymbol{B}=\sum_{j=1}^{J}\nu_j\boldsymbol{F}_j\otimes\boldsymbol{G}_j,$$

where $J=2^{(m'-m)_++(n'-n)_+}\wedge 2^{M+N-m\vee m'-n\vee n'}$ and

$$\nu_1=\|\mathcal{R}_{(m'-m)_+,(n'-n)_+}[\boldsymbol{B}]\|_S.$$

With the two KPD of $\boldsymbol{A}$ and $\boldsymbol{B}$, we can rewrite $\boldsymbol{A}\otimes\boldsymbol{B}$ as

$$\boldsymbol{A}\otimes\boldsymbol{B}=\left(\sum_{i=1}^{I}\mu_i\boldsymbol{C}_i\otimes\boldsymbol{D}_i\right)\otimes\left(\sum_{j=1}^{J}\nu_j\boldsymbol{F}_j\otimes\boldsymbol{G}_j\right)=\sum_{i=1}^{I}\sum_{j=1}^{J}\mu_i\nu_j\boldsymbol{C}_i\otimes\boldsymbol{D}_i\otimes\boldsymbol{F}_j\otimes\boldsymbol{G}_j.$$

Notice that the Kronecker product satisfies distributive law and associative law. The matrix $\boldsymbol{D}_i$ is $2^{(m-m')_+}\times 2^{(n-n')_+}$ and the matrix $\boldsymbol{F}_j$ is $2^{(m'-m)_+}\times 2^{(n'-n)_+}$. For all possible values of m,m',n,n', either one of $\boldsymbol{D}_i$ and $\boldsymbol{F}_j$ is a scalar, or they are both vectors; and for both cases $\boldsymbol{D}_i\otimes\boldsymbol{F}_j=\boldsymbol{F}_j\otimes\boldsymbol{D}_i$. Therefore,

$$\boldsymbol{A}\otimes\boldsymbol{B}=\sum_{i=1}^{I}\sum_{j=1}^{J}\mu_i\nu_j\boldsymbol{C}_i\otimes\boldsymbol{F}_j\otimes\boldsymbol{D}_i\otimes\boldsymbol{G}_j=\sum_{i=1}^{I}\sum_{j=1}^{J}\mu_i\nu_j\boldsymbol{P}_{ij}\otimes\boldsymbol{Q}_{ij},\tag{B.19}$$

where

$$\boldsymbol{P}_{ij}:=\boldsymbol{C}_i\otimes\boldsymbol{F}_j,\quad \boldsymbol{Q}_{ij}:=\boldsymbol{D}_i\otimes\boldsymbol{G}_j.$$

Notice that $\boldsymbol{P}_{ij}$ is a $2^{m'}\times 2^{n'}$ matrix and $\boldsymbol{Q}_{ij}$ is a $2^{M-m'}\times 2^{N-n'}$ matrix. Therefore, (B.19) is a KPD of $\boldsymbol{A}\otimes\boldsymbol{B}$ indexed by (i,j) with respect to the Kronecker configuration $(m',n',M-m',N-n')$ as long as $\boldsymbol{P}_{ij}$ and $\boldsymbol{Q}_{ij}$ satisfy the orthonormal condition in (B.18). In fact,

$$\begin{aligned}\langle\boldsymbol{P}_{ij},\boldsymbol{P}_{kl}\rangle&=\mathrm{tr}[\boldsymbol{P}_{ij}'\boldsymbol{P}_{kl}]\\&=\mathrm{tr}[(\boldsymbol{C}_i\otimes\boldsymbol{F}_j)'(\boldsymbol{D}_k\otimes\boldsymbol{G}_l)]\\&=\mathrm{tr}[(\boldsymbol{C}_i'\boldsymbol{D}_k)\otimes(\boldsymbol{F}_j'\boldsymbol{G}_l)]\\&=\mathrm{tr}[\boldsymbol{C}_i'\boldsymbol{D}_k]\mathrm{tr}[\boldsymbol{F}_j'\boldsymbol{G}_l]\end{aligned}$$

$$= \delta_{i,j}\delta_{k,l},$$

and similar results hold for $\boldsymbol{Q}_{ij}$. It follows that

$$\|\mathcal{R}_{m',n'}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S = \max_{i,j}\mu_i\nu_j = \mu_1\nu_1 = \|\mathcal{R}_{m\wedge m',n\wedge n'}[\boldsymbol{A}]\|_S \cdot \|\mathcal{R}_{(m'-m)_+,(n'-n)_+}[\boldsymbol{B}]\|_S,$$

and the proof of Lemma 9.1 is complete.

Now we consider Corollary 9.2. When $\boldsymbol{A}$ and $\boldsymbol{B}$ are generated as in Example 9.1, we have

$$\begin{aligned}
\|\mathcal{R}_{m\wedge m',n\wedge n'}[\tilde{\boldsymbol{A}}]\|_S &\leqslant 2^{(m\wedge m'+n\wedge n')/2} + 2^{((m-m')_+ + (n-n')_+)/2} + O_p(1),\\
\|\mathcal{R}_{(m'-m)_+,(n'-n)_+}[\tilde{\boldsymbol{B}}]\|_S &\leqslant 2^{((m'-m)_+ + (n'-n)_+)/2} + 2^{(M+N-m\vee m'-n\vee n')/2} + O_p(1),\\
\|\tilde{\boldsymbol{A}}\|_F\|\tilde{\boldsymbol{B}}\|_F &= 2^{(M+N)/2}(1+O_p(r_0)).
\end{aligned}$$

Hence,

$$\begin{aligned}
\|\mathcal{R}_{m',n'}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S = \frac{\|\mathcal{R}_{m',n'}[\tilde{\boldsymbol{A}}\otimes\tilde{\boldsymbol{B}}]\|_S}{\|\tilde{\boldsymbol{A}}\|_F\|\tilde{\boldsymbol{B}}\|_F} &\leqslant 2^{-(m'+n')/2} + 2^{-(M+N-m'-n')/2}\\
&\quad + 2^{-(|m-m'|+|n-n'|)/2} + 2^{-(M+N-|m-m'|-|n-n'|)/2} + o_p(1).
\end{aligned}$$

The maximum of the right hand side is obtained when $|m-m'|+|n-n'|=1$, or $m'+n' \in \{1, M+N-1\}$, for which

$$\|\mathcal{R}_{m',n'}[\boldsymbol{A}\otimes\boldsymbol{B}]\|_S \leqslant 1/\sqrt{2} + o_p(1).$$

Furthermore, it is straightforward to verify that the upper bound is attained when $m'+n' \in \{1, M+N-1\}$, which leads to Corollary 9.2.

Proof of Lemma 9.2

We first prove the following technical lemma.

Lemma B.3. *Let U, V be two vector subspaces of $\mathbb{R}^n$ with $\Theta(U,V) = \theta \in [0,\pi/2]$, where $\Theta(U,V)$ denotes the smallest principal angle between U and V. Suppose $w \in \mathbb{R}^n$ is a unit vector and*

$$\|P_U w\| = \cos\alpha,$$

for some $\alpha \in [0,\pi/2]$, where P_U denotes the orthogonal projection to the space U. Then it holds that

$$\|P_V w\| \leqslant \begin{cases} \cos(\theta-\alpha) & \text{if } \alpha \leqslant \theta, \\ 1 & \text{if } \alpha > \theta. \end{cases}$$

Proof. Let

$$u = \frac{P_U w}{\|P_U w\|},$$

then $\|u\| = 1$ and $u \in U$. Let $\{u_1, u_2, \ldots, u_n\}$ be an orthogonal basis of $\mathbb{R}^n$ such that $u_1 = u$. For any vector $v \in V$, we have

$$\begin{aligned} v'w &= v'\left(\sum_{i=1}^n u_i u_i'\right) w \\ &= v'u_1u_1'w + \sum_{i=2}^n v'u_iu_i'w \\ &\leqslant v'u_1u_1'w + \sqrt{\sum_{i=2}^n v'u_i}\sqrt{\sum_{i=2}^n u_i'w} \\ &= \cos\eta\cos\alpha + \sin\eta\sin\alpha \\ &= \cos(\eta-\alpha), \end{aligned}$$

where $v'u_1 = \cos\eta$. The proof is complete by noting that $\cos\eta = v'u_1 \leqslant \cos\theta$. □

We now prove Lemma 9.2.

Proof of Lemma 9.2. Recall that $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are of the same dimension. We consider the maximization of $\|(\boldsymbol{M}_1+\boldsymbol{M}_2)u\|^2$ over all unit vectors u. First write

$$\begin{aligned} \|(\boldsymbol{M}_1+\boldsymbol{M}_2)u\|^2 &= \|\boldsymbol{M}_1u + \boldsymbol{M}_2u\|^2 \\ &= \|\boldsymbol{M}_1P_{\boldsymbol{M}_1'}u + \boldsymbol{M}_2P_{\boldsymbol{M}_2'}u\|^2 \end{aligned}$$

$$= \|\boldsymbol{M}_1 P_{\boldsymbol{M}_1'} u\|^2 + \|\boldsymbol{M}_2 P_{\boldsymbol{M}_2'} u\|^2 + 2(\boldsymbol{M}_1 P_{\boldsymbol{M}_1'} u)' \boldsymbol{M}_2 P_{\boldsymbol{M}_2'} u,$$

where $P_{\boldsymbol{M}}$ denotes the projection matrix to the column space of $\boldsymbol{M}$. Since $\|\boldsymbol{M}_1\|_S = \mu$ and $\|\boldsymbol{M}_2\|_S = \nu$, we have

$$\|\boldsymbol{M}_1 P_{\boldsymbol{M}_1'} u\|^2 \leqslant \mu^2 \|P_{\boldsymbol{M}_1'} u\|^2 \quad \text{and} \quad \|\boldsymbol{M}_2 P_{\boldsymbol{M}_2'} u\|^2 \leqslant \nu^2 \|P_{\boldsymbol{M}_2'} u\|^2.$$

Since $\boldsymbol{M}_1 P_{\boldsymbol{M}_1'} u \in \operatorname{span}(\boldsymbol{M}_1)$ and $\boldsymbol{M}_2 P_{\boldsymbol{M}_2'} u \in \operatorname{span}(\boldsymbol{M}_2)$, it holds that

$$(\boldsymbol{M}_1 P_{\boldsymbol{M}_1'} u)' \boldsymbol{M}_2 P_{\boldsymbol{M}_2'} u \leqslant \cos\theta \mu\nu \|P_{\boldsymbol{M}_1'} u\| \|P_{\boldsymbol{M}_2'} u\|.$$

It follows that

$$\|(\boldsymbol{M}_1 + \boldsymbol{M}_2) u\|^2 \leqslant \mu^2 \|P_{\boldsymbol{M}_1'} u\|^2 + \nu^2 \|P_{\boldsymbol{M}_2'} u\|^2 + 2\mu\nu \|P_{\boldsymbol{M}_1'} u\| \|P_{\boldsymbol{M}_2'} u\| \cos\theta.$$

Suppose $\|P_{\boldsymbol{M}_1'} u\| = \cos\alpha$ for some $\alpha \in [0, \pi/2]$. If $\alpha > \eta$, then $\|P_{\boldsymbol{M}_2'} u\| \leqslant 1$. The right hand side of the preceding inequality attains its maximum when $\|P_{\boldsymbol{M}_1'} u\| = \cos\eta$ and $\|P_{\boldsymbol{M}_2'} u\| = 1$. Hence, we only consider the case $\alpha \leqslant \eta$, which implies that $\|P_{\boldsymbol{M}_2'} u\| \leqslant \cos(\eta - \alpha)$, and

$$\|(\boldsymbol{M}_1 + \boldsymbol{M}_2) u\|^2 \leqslant \mu^2 \cos^2\alpha + \nu^2 \cos^2(\eta - \alpha) + 2\mu\nu \cos\theta \cos\alpha \cos(\eta - \alpha).$$

Therefore,

$$\begin{aligned}
&\mu^2 \cos^2\alpha + \nu^2 \cos^2(\eta - \alpha) + 2\mu\nu \cos\theta \cos\alpha \cos(\eta - \alpha) \\
=&\frac{1}{2}\mu^2(1 + \cos 2\alpha) + \frac{1}{2}\nu^2(1 + \cos(2\eta - 2\alpha)) + \mu\nu \cos\theta[\cos\eta + \cos(\eta - 2\alpha)] \\
=&\frac{1}{2}(\mu^2 + \nu^2 + 2\mu\nu \cos\theta \cos\eta) \\
&+ \left(\frac{1}{2}\mu^2 + \frac{1}{2}\nu^2 \cos(2\eta) + \mu\nu \cos\theta \cos\eta\right) \cos(2\alpha) + \left(\frac{1}{2}\nu^2 \sin(2\eta) + \mu\nu \cos\theta \sin\eta\right) \sin(2\alpha) \\
\leqslant&\frac{1}{2}(\mu^2 + \nu^2 + 2\mu\nu \cos\theta \cos\eta) \\
&+ \sqrt{\left(\frac{1}{2}\mu^2 + \frac{1}{2}\nu^2 \cos(2\eta) + \mu\nu \cos\theta \cos\eta\right)^2 + \left(\frac{1}{2}\nu^2 \sin(2\eta) + \mu\nu \cos\theta \sin\eta\right)^2}
\end{aligned}$$

$$=\frac{1}{2}\left(\mu^2+\nu^2+2\mu\nu\cos\theta\cos\eta+\sqrt{(\mu^2+\nu^2+2\mu\nu\cos\theta\cos\eta)^2-4\mu^2\nu^2\sin^2\theta\sin^2\eta}\right).$$

The proof is complete.

Proofs of Theorem 9.5 and Corollary 9.3

The proof of Theorem 9.5 is similar to the proofs of Theorem 9.2 and Theorem 9.3, so we only point out the main steps here and omit the details.

Following the same argument as in the proof of Theorem 9.2, the expected information criteria of the true configuration is

$$\mathrm{EIC}_\kappa(m_0,n_0)=D\left[\ln\left(\lambda_2^2+\sigma^2\right)+\kappa r_0^2\right].$$

For a wrong configuration $(m,n)\in\mathcal{W}$, $\hat{\lambda}^{(m,n)}$ is obtained by

$$\hat{\lambda}^{(m,n)}=\|\lambda_1\mathcal{R}[\boldsymbol{A}_1\otimes\boldsymbol{B}_1]+\lambda_2\mathcal{R}[\boldsymbol{A}_2\otimes\boldsymbol{B}_2]+\sigma D^{-1/2}\mathcal{R}[\boldsymbol{E}]\|_S.$$

According to Lemma 9.2 and Assumption 9.5, we have

$$\|\lambda_1\mathcal{R}[\boldsymbol{A}_1\otimes\boldsymbol{B}_1]+\lambda_2\mathcal{R}[\boldsymbol{A}_2\otimes\boldsymbol{B}_2]\|_S^2\leqslant\lambda_1^2\phi_1^2+\lambda_2^2\phi_2^2+2\lambda_1\lambda_2\phi_1\phi_2\xi<(\lambda_1+\lambda_2)^2. \tag{B.20}$$

By Lemma B.2, we have

$$[\hat{\lambda}^{(m,n)}]^2\leqslant\lambda_1^2\phi_1^2+\lambda_2^2\phi_2^2+2\lambda_1\lambda_2\phi_1\phi_2\xi+\sigma^2r_{m,n}^2 +O((\lambda_1+\lambda_2)\sigma D^{-1/4})+O_p\left((\lambda_1+\lambda_2+\sigma)\sigma D^{-1/2}\right). \tag{B.21}$$

With (B.21) replacing (B.4), the rest of the proof follows the same line of the proof of Theorem 9.2.

The proof of consistency is same as in the proof of Theorem 9.3 except that the formula of $\hat{\lambda}^{(m,n)}$ in (B.21) is used in (B.15).

We now prove Corollary 9.3. When model (9.17) is generated under the random scheme

in Example 9.2, we only consider the wrong configuration close to the true configuration. It can be verified that the separation $\Delta\text{EIC}(m,n)$ is larger at other configurations. Consider (m,n) such that $|m_0-m|+|n_0-n|=1$. Then from Corollary 9.2, we have

$$\phi_1=\frac{1}{\sqrt{2}}+O_p(r_0),\quad \phi_2=\frac{1}{\sqrt{2}}+O_p(r_0).$$

Now consider the principle angles between $\mathcal{R}[\boldsymbol{A}_1\otimes\boldsymbol{B}_1]$ and $\mathcal{R}[\boldsymbol{A}_2\otimes\boldsymbol{B}_2$ as in Lemma 9.2, We have

$$\cos\theta=O_p(2^{-(m+n)}),\quad \cos\eta=O_p(2^{-(m^\dagger+n^\dagger)}).$$

By Lemma 9.2, (B.20) can be revised to

$$\|\lambda_1\mathcal{R}[\boldsymbol{A}_1\otimes\boldsymbol{B}_1]+\lambda_2\mathcal{R}[\boldsymbol{A}_2\otimes\boldsymbol{B}_2]\|_S^2\leqslant\frac{\lambda_1^2}{2}+O_p(\lambda_1^2 r_0).$$

Corollary 9.3 follows immediately.