DIRECTED LEARNING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Yi-Hao Kao
June 2012

This dissertation is online at: http://purl.stanford.edu/bq257xh0795

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Benjamin Van Roy, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Stephen Boyd, Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Trevor Hastie

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

In machine learning, it is common to treat estimation of model parameters separately from subsequent use of the model to guide decisions. In particular, the learning process typically aims to maximize "goodness of fit" without consideration of decision objectives. In this dissertation, we propose a new approach – *directed learning* – which factors decision objectives into the model fitting procedure in order to improve decision quality. We develop and analyze directed learning algorithms for three classes of problems. In the first case, we consider a problem where linear regression analysis is used to guide decision making. We propose *directed regression*, an efficient algorithm that takes into account the decision objective when computing regression coefficients. We demonstrate through a computational study that directed regression can generate significant performance gains, and establish a theoretical result that motivates it. This setting is then extended to a multi-stage decision problem as our second case, and we show that a variation of directed regression, *directed time-series regression*, improves performance in this context as well. Lastly, we consider a problem that involves estimating a covariance matrix and making a decision based on that estimate. Such problems arise in portfolio management among other areas, and a common approach is to employ principal component analysis (PCA) to estimate a parsimonious factor model. We propose *directed PCA*, an efficient algorithm that accounts for the decision objective in the selection of components, and demonstrate through experiments that it leads to significant improvement. We also establish through a theoretical result that the possible degree of improvement can be unbounded.

To my parents – Chi-Shan Kao and Su-Chen Lee

Acknowledgments

This thesis work would not be possible without the guidance from my principal advisor, Professor Benjamin Van Roy. In addition to the technical knowledge he taught me over the past five years, I especially appreciate the solid attitude he demonstrated to me through approaching research problems and presenting results. The thoroughness in his thinking process, and the clarity in his articulation, are the best virtues I have ever learned through my PhD study at Stanford. Furthermore, his willingness to share his experiences in both academia and various industries has always been a bliss for me, and greatly broadened my horizons.

I am also indebted to my associate advisor, Professor Stephen Boyd. Besides teaching me dynamical systems and convex optimization, his comments and suggestions on this work have been a precious input, from which I benefit extensively.

In addition, I am deeply grateful to my oral committee members Professor Andrea Montanari and Professor Mohsen Bayati, and my reading committee member Professor Trevor Hastie. Their remarkable insights have helped me develop different perspectives on this research, and have much contributed to its improvement.

Part of this research was motivated by industry applications. For that, I would like to thank my colleagues during my internships at Yahoo! Labs and Goldman Sachs, including Alexander Smola, Emre Velipasaoglu, and Marty Young. Working with them has inspired me to formulate and tackle research problems from a practical point of view.

I feel privileged to be surrounded by my fellow group members Xiang Yan, Michael Padilla, Beomsoo Park, Zheng Wen, Waraporn Tongprasit, and Hamid Reza Maei. I could not overstate how much I enjoy and benefit from our interactions. Among

them, special thanks go to Xiang Yan, who introduced me to this research group and served as my mentor during the very early stage of my research career.

I would also like to thank the faculty members from my research community who have granted very helpful discussions with me over the years. These include Professor Teresa Meng, Professor Thomas Dietterich, Professor Ciamac Moallemi, Professor Daniel Rubin, Professor Sandy Napel, Professor Andrew Ng, Professor Daphne Koller, Professor Shie Mannor, Professor Robert Tibshirani, and Professor Vivek Farias.

Finally, I would like to thank my friends and family for your support. Your companionship is always encouraging, and it made me who I am today.

The author was supported by a Stanford Graduate Fellowship. This work was supported in part by the National Science Foundation through grant CMMI-0653876 and CMMI-0968707.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Applications of machine learning in which estimated models are used to guide decisions are ubiquitous: customer demand models are learned and used to optimize inventory control decisions; models of stock market behavior are learned to guide trade decisions; and healthcare providers learn hospitalization patterns to facilitate preventive care decisions. Nevertheless, it is common to treat estimation of model parameters separately from subsequent use of the model to guide decisions. In particular, the learning process typically aims to maximize "goodness of fit" without consideration of decision objectives. In this dissertation, we propose a new approach – directed learning – which factors decision objectives into the model fitting procedure in order to improve decision quality.

1.1 Overview

We develop and analyze directed learning algorithms for three classes of problems in Chapter 2, 3, and 4, respectively. The first two cases focuses on supervised learning problems, whereas the last one unsupervised.

In Chapter 2, we consider a scenario where linear regression analysis is used to guide decision making. We propose *directed regression*, an efficient algorithm that takes into account the decision objective when computing regression coefficients. We demonstrate through a computational study that directed regression can generate

significant performance gains over conventional methods. We also develop a Bayesian framework in which we establish a theoretical result that motivates this algorithm.

We then extend the application of directed regression from a single-stage decision problem to a multi-stage stochastic control problem in Chapter 3. We propose *directed time-series regression*, a new approach to estimating parameters of time-series models for use in certainty equivalent model predictive control. Through a computational study involving a stochastic version of a well known inverted pendulum balancing problem, we demonstrate that directed time-series regression can generate significant improvements in controller performance over conventional methods.

Finally, in Chapter 4 we consider a problem that involves estimating a covariance matrix and making a decision based on that estimate. Such problems arise in portfolio management among other areas, and a common approach is to employ principal component analysis (PCA) to estimate a parsimonious factor model. We propose *directed PCA*, an efficient algorithm that accounts for the decision objective in the selection of components, and demonstrate through experiments that it leads to significant improvement. We also establish through a theoretical result that the possible degree of improvement can be unbounded.

1.2 Main Concept

The three cases we study in this thesis revolve around a central problem: given a class of statistical models, we would like to estimate model parameters from random samples and then make a decision based on this estimate. The decision payoff depends on the decision and an additional data sample, observed after the decision is made and generated independently according to the same distribution as those previously observed. One common approach is to learn the model parameters via maximizing the goodness of fit, and then optimize the decision based on the resulting model. In practice, the goodness of fit is usually measured by the likelihood of model parameters, and therefore maximum-likelihood estimation is adopted. If we further have a prior belief about these parameters, this turns into *maximum a posteriori* (MAP) estimation.

Apparently, MAP estimation does not take into account the decision objective when producing an estimate. An alternative approach that does so is *empirical optimization* (EO), as studied for example in Bartlett and Mendelson (2006) and Haussler (1992). Let us assume for the purpose of this introduction that we are restricted to select model parameters from a set S. In empirical optimization, one assumes that the future data sample is drawn uniformly from the set of previously observed samples, and the model parameters are selected from S to maximizes expected payoff of the resulting decision. Equivalently, one can view empirical optimization as selecting from S the model parameters that would lead to a decision strategy that optimizes "in-sample performance." As such, empirical optimization does not explicitly aim to select model parameters that explains historical data; the focus is on historical performance of hypothetical decisions. Unfortunately, this approach often leads to poor out-of-sample performance. The problem is over-fitting; the selected parameters are too specialized to decisions that would have been effective in the face of previously observed data samples, and the performance of the resulting decision strategy does not generalize well to future samples.

We can view MAP and EO as two extreme approaches; the former focuses on fitting the model, whereas the latter focuses on the decision objective. Directed learning aims to seek an optimal trade-off between these two extremes. Indeed, directed regression and directed PCA are both designed with this spirit, but adopt different methodologies. Specifically, directed regression takes a convex combination of the model parameters produced by MAP and EO, whereas directed PCA essentially carries out empirical optimization but subject to a constraint that the resulting estimate is selected from a confidence region around the MAP estimate. As we shall see in the following chapters, both methodologies offer significant improvements over MAP or EO alone.

1.3 Literature Review

A conceptual thrust in the development of directed learning is in the use of a decision objective to guide model-fitting. In order to put our work in perspective relative to

prior literature, we will discuss here three threads of research that relate to this spirit: robust optimization, operational statistics, and statistical decision theory.

In robust optimization, one makes a decision assuming that uncertain parameters are chosen adversarially, though constrained to a prescribed confidence region. For example, in the context of linear regression, such confidence region is often chosen as an ellipsoid (Ben-Tal and Nemirovski, 1998). In the context of covariance matrix estimation, the work from this area that most closely relates to ours is that of Goldfarb and Iyengar (2003), which treated a robust portfolio selection problem with confidence regions for expected returns and covariances. This approach selects a portfolio that maximizes risk-adjusted expected return, assuming worst case point estimates within the confidence regions. Also related is the subsequent work of Delage and Ye (2008), which treated a more general formulation that accommodates uncertainty beyond expectations and covariances; this formulation synthesizes those of Goldfarb and Iyengar (2003) and Popescu (2007). For a detailed survey of robust optimization, the reader is referred to Bertsimas and Thiele (2006) and Ben-Tal et al. (2009).

Similarly with robust optimization, directed PCA constrains the choice of point estimate to a confidence region. However, an important difference is that, instead of selecting a worst case point estimate, our approach selects one that optimizes in-sample performance. Relative to using the MAP estimate, our approach is in a sense more aggressive whereas robust optimization is more conservative. Being aggressive helps when the MAP estimate exhibits significant bias and reasonably low variance. Bias can stem from model misspecification or MAP estimation. For example, empirical data may not be generated by a factor model with a small number of factors, and even if they were, the MAP estimate can be different from the conditional expectation. On the other hand, assuming a factor model structure controls the variance of the resulting estimate so that it is sufficiently robust without imposing any additional conservatism. Indeed, in the course of our research, we tried robust optimization using our confidence regions but found resulting decisions to be overly conservative and to perform worse than MAP estimates.

The above reasoning also applies to our settings for directed regression and directed time-series regression. In those cases, we consider misspecified regression models, and the number of regression coefficients is relatively small compared to the amount of observed data. As a consequence, the MAP estimate tends to have significant bias but reasonably low variance, which suggests nudging that estimate towards EO estimate can improve performance.

Operational statistics is another line of work that emphasizes the relevance of decision objectives in estimating model parameters. This terminology first appeared in Liyanage and Shanthikumar (2005), where the authors describe a method that factors objectives into how demand distributions are estimated when they are to be used in a newsvendor inventory control problem. In particular, instead of estimating the parameters of the distribution, the method estimates an optimal order quantity directly from the data. This approach is subsequently elaborated by Chua et al. (2008) using Bayesian analysis with a non-informative prior. In a related vein, Besbes et al. (2010) develop a statistical test that incorporates decision performance into a measure of statistical validity and illustrate their approach in the context of a revenue management problem. Our work can be seen as contributing to this line of research by developing methods that are similar in spirit but designed for different classes of problems.

Our work also relates to the broad and well-studied area of statistical decision theory (see, e.g., Berger (1985)). In this area, it is common to apply a Bayesian approach, which begins with a prior distribution over all possible models and then evaluates a posterior distribution conditioned on observed data. The Bayes optimal decision is then taken to be the one that maximizes expected payoff with respect to the posterior distribution. This approach is often computationally intractable for relevant decision objectives and model classes, and as a consequence practitioners tend to appeal to approximate solutions. In our context, MAP estimation provides an approximate solution, which generally suffers from two disadvantages. First, the prior distribution is usually chosen in a way that not only reflects our assumption but also enables efficient computation of the MAP estimate. Such mathematical convenience is often attained at the price of introducing model bias. Second, MAP estimation

does not take into account the decision objective when computing an estimate. As we shall see, these disadvantages together leave room for improvement.

Chapter 2

Directed Regression

When used to guide decisions, linear regression analysis typically involves estimation of regression coefficients via ordinary least squares and their subsequent use to make decisions. When there are multiple response variables and features do not perfectly capture their relationships, it is beneficial to account for the decision objective when computing regression coefficients. Empirical optimization does so but sacrifices performance when features are well-chosen or training data are insufficient. In this chapter, we propose *directed regression*, an efficient algorithm that combines merits of ordinary least squares and empirical optimization. We demonstrate through a computational study that directed regression can generate significant performance gains over either alternative. We also develop a theory that motivates the algorithm.

2.1 Overview

When used to guide decision-making, linear regression analysis typically treats estimation of regression coefficients separately from their use to make decisions. In particular, estimation is carried out via ordinary least squares (OLS) without consideration of the decision objective. For a Gaussian model with a non-informative prior distribution for model coefficients, OLS essentially produces an MAP estimate. We will use OLS and MAP interchangeably in this and the next chapter. After this estimation procedure, the regression coefficients are then used to optimize decisions.

When there are multiple response variables and features do not perfectly capture their relationships, it is beneficial to account for the decision objective when computing regression coefficients. Imperfections in feature selection are common since it is difficult to identify the right features and the number of features is typically restricted in order to avoid over-fitting.

Empirical optimization (EO) is an alternative to OLS which selects coefficients that minimize empirical loss in the training data. Though it accounts for the decision objective when computing regression coefficients, EO sacrifices performance when features are well-chosen or training data is insufficient.

In this chapter, we propose a new algorithm – directed regression (DR) – which is a hybrid between OLS and EO. DR selects coefficients that are a convex combination of those that would be selected by OLS and those by EO. The weights of OLS and EO coefficients are optimized via cross-validation.

We study DR for the case of decision problems with quadratic objective functions. The algorithm takes as input a training set of data pairs, each consisting of feature vectors and response variables, together with a quadratic loss function that depends on decision variables and response variables. Regression coefficients are computed for subsequent use in decision-making. Each future decision depends on newly sampled feature vectors and is made prior to observing response variables with the goal of minimizing expected loss.

We present computational results demonstrating that DR can substantially outperform both OLS and EO. These results are for synthetic problems with regression models that include subsets of relevant features. In some cases, OLS and EO deliver comparable performance while DR reduces expected loss by about 20%. In none of the cases considered does either OLS or EO outperform DR.

We also develop a theory that motivates DR. This theory is based on a model in which selected features do not perfectly capture relationships among response variables. We prove that, for this model, the optimal vector of coefficients is a convex combination of those that would be generated by OLS and EO.

For the convenience of the reader, we summarize our mathematical notation in Table 2.1. Part of this chapter has been published in Kao et al. (2009).

Notation	Definition
$\mathbf{X}^{(n)}$	The matrix consisting of feature vectors of the n-th sample
$\mathbf{y}^{(n)}$	Response vector of the n-th sample
$\mathbf{u}$	Decision vector
$\ell(\mathbf{u}, \mathbf{y})$	Decision loss function
$\mathbf{G}_1$	Quadratic coefficient of loss function
$\mathbf{G}_2$	Linear coefficient of loss function
$\mathbf{r}$	Regression coefficients vector
$\mathcal{O}$	Dataset
$\overline{\mathcal{O}}$	Augmented dataset
N	Number of sample data
M	Dimension of data
K	Number of observed features
J	Number of unobserved features
ϕ	Information vector
$\mathbf{w}$	Noise vector
σ_r	Standard deviation of the prior for observed features
σ_ϵ	Standard deviation of the prior for unobserved features
σ_w	Standard deviation of noise

Table 2.1: Notation for Chapter 2.

2.2 Linear Regression for Decision-Making

Suppose we are given a set of training data pairs $\mathcal{O} = \{(\mathbf{X}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{X}^{(N)}, \mathbf{y}^{(N)})\}$. Each nth data pair is comprised of feature vectors $\mathbf{X}^{(n)} \in \mathbb{R}^{M \times K}$ and a vector $\mathbf{y}^{(n)} \in \mathbb{R}^M$ of response variables. We would like to compute regression coefficients $\mathbf{r} \in \mathbb{R}^K$ so that given a data pair $(\mathbf{X}, \mathbf{y})$, the linear combination $\mathbf{Xr}$ of feature vectors estimates the expectation of $\mathbf{y}$ conditioned on $\mathbf{X}$. We restrict attention to cases where $M > 1$, with special interest in problems where M is large, because it is in such situations that DR offers the largest performance gains.

We consider a setting where the regression model is used to guide future decisions. In particular, after computing regression coefficients, each time we observe feature vectors $\mathbf{X}$ we will have to select a decision $\mathbf{u} \in \mathbb{R}^L$ before observing the response vector $\mathbf{y}$. The choice incurs a loss

$$\ell(\mathbf{u}, \mathbf{y}) = \mathbf{u}^{\mathrm{T}}\mathbf{G}_1\mathbf{u} + \mathbf{u}^{\mathrm{T}}\mathbf{G}_2\mathbf{y},$$

where the matrices $\mathbf{G}_1 \in \mathbb{R}^{L \times L}$ and $\mathbf{G}_2 \in \mathbb{R}^{L \times M}$ are known, and the former is positive definite and symmetric. We aim to minimize expected loss, assuming that the conditional expectation of $\mathbf{y}$ given $\mathbf{X}$ is $\mathbf{Xr}$. As such, given $\mathbf{X}$ and $\mathbf{r}$, we select a decision

$$\mathbf{u}_{\mathbf{r}}(\mathbf{X}) = \operatorname*{argmin}_u \ell\left(\mathbf{u}, \mathbf{Xr}\right) = -\frac{1}{2}\mathbf{G}_1^{-1}\mathbf{G}_2\mathbf{Xr}.$$

The question is how best to compute the regression coefficients $\mathbf{r}$ for this purpose.

To motivate the setting we have described, we offer a hypothetical application.

Example 2.1. *Consider an Internet banner ad campaign that targets M classes of customers. An average revenue of $\mathbf{y}_m$ is received per customer of class m that the campaign reaches. This quantity is random and influenced by K observable factors $\mathbf{X}_{m,1}, \ldots, \mathbf{X}_{m,K}$. These factors may be correlated across customers classes; for example, they could capture customer preferences as they relate to ad content or how current economic conditions affect customers. For each mth class, the cost of reaching the $\mathbf{u}_m$th customer increases with $\mathbf{u}_m$ because ads are first targeted at customers that can be reached at lower cost. This cost is quadratic, so that we pay $\gamma_m\mathbf{u}_m^2$ to reach*

$\mathbf{u}_m$ *customers, where* γ_m *is a known constant.*

The application we have described fits our general problem context. It is natural to predict the response vector $\mathbf{y}$ *using a linear combination* $\mathbf{Xr}$ *of factors with the regression coefficients* $\mathbf{r}$ *computed based on past observations* $\mathcal{O} = \{(\mathbf{X}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{X}^{(N)}, \mathbf{y}^{(N)})\}$*. The goal is to maximize expected revenue less advertising costs. This gives rise to a loss function that is quadratic in* $\mathbf{u}$ *and* $\mathbf{y}$*:*

$$\ell(\mathbf{u}, \mathbf{y}) = \sum_{m=1}^{M} (\gamma_m \mathbf{u}_m^2 - \mathbf{u}_m \mathbf{y}_m).$$

One might ask why not construct M separate linear regression models, one for each response variable, each with a separate set of K coefficients. The reason is that this gives rise to MK coefficients; when M is large and data is limited, this could lead to over-fitting. Models of the sort we consider, where regression coefficients are shared across multiple response variables, are sometimes referred to as *general linear models* and have seen a wide range of applications (Kim and Timm, 2006; Muller and Stewart, 2006). It is well-known that the quality of results is highly sensitive to the choice of features, even more so than for models involving a single response variable (Kim and Timm, 2006).

2.3 Algorithms

Ordinary least squares (OLS) is a conventional approach to computing regression coefficients. This would produce a coefficient vector

$$\mathbf{r}^{\mathrm{OLS}} = \operatorname*{argmin}_{\mathbf{r} \in \mathbb{R}^K} \sum_{n=1}^{N} \left\| \mathbf{y}^{(n)} - \mathbf{X}^{(n)} \mathbf{r} \right\|^2. \tag{2.1}$$

Note that OLS does not take the decision objective into account when computing regression coefficients. Empirical optimization (EO) offers an alternative that does

so. This approach minimizes empirical loss on the training data:

$$\mathbf{r}^{\mathrm{EO}} = \operatorname*{argmin}_{\mathbf{r}\in\mathbb{R}^K} \sum_{n=1}^{N} \ell(\mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)}), \mathbf{y}^{(n)}). \tag{2.2}$$

Note that EO does not explicitly aim to estimate the conditional expectation of the response vector. Instead it focusses on decision loss that would be incurred with the training data. Both $\mathbf{r}^{\mathrm{OLS}}$ and $\mathbf{r}^{\mathrm{EO}}$ can be computed efficiently by minimizing convex quadratic functions.

As we will see in our computational and theoretical analyses, OLS and EO can be viewed as two extremes, each offering room for improvement. In this chapter, we propose an alternative algorithm – directed regression (DR) – which produces a convex combination $\mathbf{r}^{\mathrm{DR}} = (1-\lambda)\mathbf{r}^{\mathrm{OLS}} + \lambda\mathbf{r}^{\mathrm{EO}}$ of coefficients computed by OLS and EO. The term *directed* is chosen to indicate that DR is influenced by the decision objective though, unlike EO, it does not simply minimize empirical loss. The parameter $\lambda \in [0, 1]$ is computed via cross-validation, with an objective of minimizing average loss on validation data. Average loss is a convex quadratic function of λ, and therefore can be easily minimized over $\lambda \in [0, 1]$.

2.4 Computational Results

In this section, we present results from applying OLS, EO, and DR to synthetic data. To generate a data set, we first sample parameters of a generative model as follows:

1. Sample P matrices $\mathbf{C}_1, \ldots, \mathbf{C}_P \in \mathbb{R}^{M\times Q}$, with each entry from each matrix drawn independently from $\mathcal{N}(0, 1)$.
2. Sample a vector $\tilde{\mathbf{r}} \in \mathbb{R}^P$ from $\mathcal{N}(0, \mathbf{I})$.
3. Sample $\mathbf{G}_a \in \mathbb{R}^{L\times L}$ and $\mathbf{G}_b \in \mathbb{R}^{L\times M}$, with each entry of each matrix drawn from $\mathcal{N}(0, 1)$. Let $\mathbf{G}_1 = \mathbf{G}_a^{\mathrm{T}}\mathbf{G}_a$ and $\mathbf{G}_2 = \mathbf{G}_a^{\mathrm{T}}\mathbf{G}_b$.

Given generative model parameters $\mathbf{C}_1, \ldots, \mathbf{C}_P$ and $\tilde{\mathbf{r}}$, we sample each training data pair $(\mathbf{X}^{(n)}, \mathbf{y}^{(n)})$ as follows:

1. Sample a vector $\phi^{(n)} \in \mathbb{R}^Q$ from $\mathcal{N}(0, \mathbf{I})$ and a vector $\mathbf{w}^{(n)} \in \mathbb{R}^M$ from $\mathcal{N}(0, \sigma_w^2\mathbf{I})$.

2. Let $\mathbf{y}^{(n)} = \sum_{i=1}^{P} \tilde{\mathbf{r}}_i \mathbf{C}_i \phi^{(n)} + \mathbf{w}^{(n)}$.
3. Let $\mathbf{X}^{(n)} = \begin{bmatrix} \mathbf{C}_1\phi^{(n)} & \dots & \mathbf{C}_K\phi^{(n)} \end{bmatrix}$.

The vector $\phi^{(n)}$ can be viewed as a sample from an underlying information space. The matrices $\mathbf{C}_1, \dots, \mathbf{C}_P$ extract feature vectors from $\phi^{(n)}$. Note that, though response variables depend on P feature vectors, only $K \leq P$ are used in the regression model.

Given generative model parameters and a coefficient vector $\mathbf{r} \in \mathbb{R}^K$, it is easy to evaluate the expected loss $\overline{\ell}(\mathbf{r}) = \mathrm{E}_{\mathbf{X},\mathbf{y}}[\ell(\mathbf{u}_\mathbf{r}(\mathbf{X}), \mathbf{y})]$. It is also easy to evaluate the minimal expected loss $\ell^* = \min_r \mathrm{E}_{\mathbf{X},\mathbf{y}}[\ell(\mathbf{u}_\mathbf{r}(\mathbf{X}), \mathbf{y})]$. We will assess each algorithm in terms of the excess loss $\overline{\ell}(\mathbf{r}) - \ell^*$ delivered by the coefficient vector $\mathbf{r}$ that the algorithm computes. Excess loss is nonnegative, and this allows us to make comparisons in percentage terms.

We carried out two sets of experiments to compare the performance of OLS, EO, and DR. In the first set, we let $M = 15$, $L = 15$, $P = 60$, $Q = 20$, $\sigma_w = 5$, and $K = 50$. For each $N \in \{10, 15, 20, 30, 50\}$, we ran 100 trials, each with an independently sampled generative model and training data set. In each trial, each algorithm computes a coefficient vector given the training data and loss function. With DR, λ is selected via leave-one-out cross-validation when $N \leq 20$, and via 5-fold cross-validation when $N > 20$. Figure 2.1(a) plots excess losses averaged over trials. Note that the excess loss incurred by DR is never larger than that of OLS or EO. Further, when $N = 20$, the excess loss of OLS and EO are both around 20% larger than that of DR. For small N, OLS is as effective as DR, while EO becomes as effective as DR as N grows large.

In the second set of experiments, we use the same parameter values as in the first set, except we fix $N = 20$ and consider use of $K \in \{45, 50, 55, 58, 60\}$ feature vectors. Again, we ran 100 trials for each K, applying the three algorithms as in the first set of experiments. Figure 2.1(b) plots excess losses averaged over trials. Note that when $K = 55$, DR delivers excess loss around 20% less than EO and OLS. When $K = P = 60$, there are no missing features and OLS matches the performance of DR.

Figure 2.2 plots the values of λ selected by cross-validation, each averaged over the 100 trials, as a function of N and K. As the number of training samples N grows, so does λ, indicating that DR is weighted more heavily toward EO. As the number of

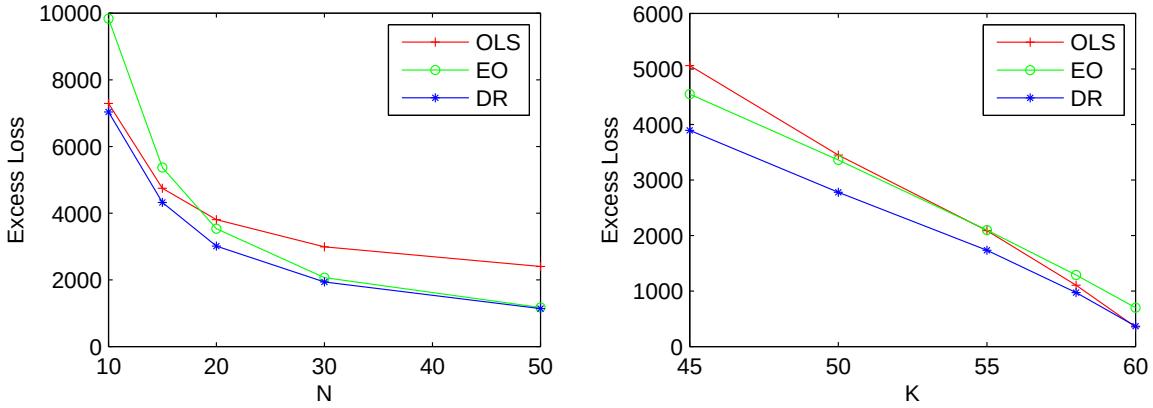


Figure 2.1: (a) Excess losses delivered by OLS, EO, and DR, for different numbers N of training samples. (b) Excess losses delivered by OLS, EO, and DR, using different numbers K of the 60 features.

feature vectors K grows, λ diminishes, indicating that DR is weighted more heavily toward OLS.

2.5 Theoretical Analysis

In this section, we formulate a generative model for the training data and future observations. For this model, optimal coefficients are convex combinations of $\mathbf{r}^{\mathrm{OLS}}$ and $\mathbf{r}^{\mathrm{EO}}$. As such, our model and analysis motivate the use of DR.

2.5.1 Model

In this section, we describe a generative model that samples the training data set, as well as "missing features," and a representative future observation. We then formulate an optimization problem where the objective is to minimize expected loss on the future observation conditioned on the training data and missing features. It may seem strange to condition on missing features since in practice they are unavailable when computing regression coefficients. However, we will later establish that optimal coefficients are convex combinations of $\mathbf{r}^{\mathrm{OLS}}$ and $\mathbf{r}^{\mathrm{EO}}$, each of which can be computed

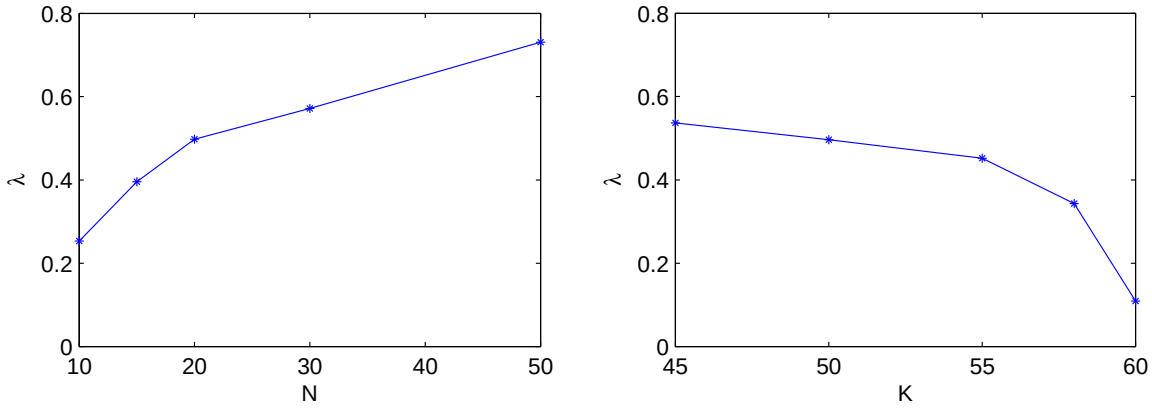


Figure 2.2: (a) The average values of selected λ, for different numbers N of training samples. (b) The average values of selected λ, using different numbers K of the 60 features.

without observing missing features. Since directed regression searches over these convex combinations, it should approximate what would be generated by a hypothetical algorithm that observes missing features.

We will assume that each feature, whether observed or missing, is a linear function of an "information vector" drawn from $\mathbb{R}^Q$. Specifically, the N training data samples depend on information vectors $\phi^{(1)}, \ldots, \phi^{(N)} \in \mathbb{R}^Q$. A linear function mapping an information vector to a feature vector can be represented by a matrix in $\mathbb{R}^{M \times Q}$, and to describe our generative model, it is useful to define an inner product for such matrices. In particular, we define the inner product between matrices $\mathbf{A}$ and $\mathbf{B}$ by

$$\langle \mathbf{A}, \mathbf{B} \rangle = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{A}\phi^{(n)})^{\mathrm{T}} (\mathbf{B}\phi^{(n)}).$$

Our generative model takes several parameters as input. First, there are the number of samples N, the number of response variables M, and the number of feature vectors K. Second, a parameter μ_Q specifies the expected dimension of the information vector. Finally, there are standard deviations σ_r, σ_ϵ, and σ_w, of observed feature coefficients, missing feature coefficients, and noise, respectively. Given parameters N,

M, K, μ_Q, σ_r, σ_ϵ, and σ_w, the generative model produces data as follows:

1. Sample Q from the geometric distribution with mean μ_Q.
2. Sample $\phi^{(1)}, \ldots, \phi^{(N)} \in \mathbb{R}^Q$ from $\mathcal{N}(0, \mathbf{I}_Q)$.
3. Sample $\mathbf{C}_1, \ldots, \mathbf{C}_K$ and $\mathbf{D}_1, \cdots, \mathbf{D}_J \in \mathbb{R}^{M \times Q}$ with each entry i.i.d. from $\mathcal{N}(0, 1)$, where $K + J = MQ$.
4. Apply the Gram-Schmidt algorithm with respect to the inner product defined above to generate an orthonormal basis $\tilde{\mathbf{C}}_1, \ldots, \tilde{\mathbf{C}}_K, \tilde{\mathbf{D}}_1, \ldots, \tilde{\mathbf{D}}_J$ from the sequence $\mathbf{C}_1, \ldots, \mathbf{C}_K, \mathbf{D}_1, \ldots, \mathbf{D}_J$.
5. Sample $\mathbf{r}^* \in \mathbb{R}^K$ from $\mathcal{N}(0, \sigma_r^2 \mathbf{I}_K)$ and $\mathbf{r}^{\perp *} \in \mathbb{R}^J$ from $\mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I}_J)$.
6. For $n = 1, \cdots, N$, sample $\mathbf{w}^{(n)} \in \mathbb{R}^M$ from $\mathcal{N}(0, \sigma_w^2 \mathbf{I}_M)$, and let

$$\mathbf{X}^{(n)} = \begin{bmatrix} \mathbf{C}_1 \phi^{(n)} & \cdots & \mathbf{C}_K \phi^{(n)} \end{bmatrix}, \tag{2.3}$$

$$\mathbf{Z}^{(n)} = \begin{bmatrix} \tilde{\mathbf{D}}_1 \phi^{(n)} & \cdots & \tilde{\mathbf{D}}_J \phi^{(n)} \end{bmatrix}, \tag{2.4}$$

$$\mathbf{y}^{(n)} = \mathbf{X}^{(n)} \mathbf{r}^* + \mathbf{Z}^{(n)} \mathbf{r}^{\perp *} + \mathbf{w}^{(n)}. \tag{2.5}$$

7. Sample $\tilde{\phi}$ uniformly from $\{\phi^{(1)}, \cdots, \phi^{(N)}\}$ and $\tilde{\mathbf{w}} \in \mathbb{R}^M$ from $\mathcal{N}(0, \sigma_w^2 \mathbf{I}_M)$. Generate $\tilde{\mathbf{X}}$, $\tilde{\mathbf{Z}}$, and $\tilde{\mathbf{y}}$ by the same functions in (2.3), (2.4), and (2.5).

The samples $\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(N)}, \tilde{\mathbf{Z}}$ represent missing features. The Gram-Schmidt procedure ensures two properties. First, since $\langle \mathbf{C}_k, \tilde{\mathbf{D}}_j \rangle = 0$, missing features are uncorrelated with observed features. If this were not the case, observed features would provide information about missing features. Second, since $\tilde{\mathbf{D}}_1, \ldots, \tilde{\mathbf{D}}_J$ are orthonormal, the distribution of missing features is invariant to rotations in the J-dimensional subspace from which they are drawn. In other words, all directions in that space are equally likely.

We define an augmented training set $\overline{\mathcal{O}} = \{(\mathbf{X}^{(1)}, \mathbf{Z}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{X}^{(N)}, \mathbf{Z}^{(N)}, \mathbf{y}^{(N)})\}$ and consider selecting regression coefficients $\hat{\mathbf{r}} \in \mathbb{R}^K$ that solve

$$\min_{\mathbf{r} \in \mathbb{R}^K} \mathrm{E}[\ell(\mathbf{u}_\mathbf{r}(\tilde{\mathbf{X}}), \tilde{\mathbf{y}}) | \overline{\mathcal{O}}].$$

Note that the probability distribution here is implicitly defined by our generative model, and as such, $\hat{\mathbf{r}}$ may depend on N, M, K , μ_Q, σ_r, σ_ϵ, σ_w, and $\overline{\mathcal{O}}$.

2.5.2 Optimal Solutions

Our primary interest is in cases where prior knowledge about the coefficients $\mathbf{r}^*$ is weak and does not significantly influence $\hat{\mathbf{r}}$. As such, we will from here on restrict attention to the case where σ_r is asymptotically large. Hence, $\hat{\mathbf{r}}$ will no longer depend on σ_r.

It is helpful to consider two special cases. One is where $\sigma_\epsilon = 0$ and the other is where σ_ϵ is asymptotically large. We will refer to $\hat{\mathbf{r}}$ in these extreme cases as $\hat{\mathbf{r}}_0$ and $\hat{\mathbf{r}}_\infty$. The following theorem establishes that these extremes are delivered by OLS and EO.

Theorem 2.1. *For all N, M, K, μ_Q, σ_w, and $\overline{\mathcal{O}}$,*

$$\hat{\mathbf{r}}_0 = \operatorname*{argmin}_{\mathbf{r}\in\mathbb{R}^K} \sum_{n=1}^{N} \left\|\mathbf{y}^{(n)} - \mathbf{X}^{(n)}\mathbf{r}\right\|^2$$

and

$$\hat{\mathbf{r}}_\infty = \operatorname*{argmin}_{\mathbf{r}\in\mathbb{R}^K} \sum_{n=1}^{N} \ell(\mathbf{u}_\mathbf{r}(\mathbf{X}^{(n)}), \mathbf{y}^{(n)}).$$

Note that σ_ϵ represents the degree of bias in a regression model that assumes there are no missing features. Hence, the above theorem indicates that OLS is optimal when there is no bias while EO is optimal as the bias becomes asymptotically large. It is also worth noting that the coefficient vectors $\hat{\mathbf{r}}_0$ and $\hat{\mathbf{r}}_\infty$ can be computed without observing the missing features, though $\hat{\mathbf{r}}$ is defined by an expectation that is conditioned on their realizations. Further, computation of $\hat{\mathbf{r}}_0$ and $\hat{\mathbf{r}}_\infty$ does not require knowledge of Q or σ_w.

Our next theorem establishes that the coefficient vector $\hat{\mathbf{r}}$ is always a convex combination of $\hat{\mathbf{r}}_0$ and $\hat{\mathbf{r}}_\infty$.

Theorem 2.2. *For all N, M, K, μ_Q, σ_w, σ_ϵ, and $\overline{\mathcal{O}}$,*

$$\hat{\mathbf{r}} = (1-\lambda)\hat{\mathbf{r}}_0 + \lambda\hat{\mathbf{r}}_\infty,$$

where $\lambda = \frac{N\sigma_\epsilon^2}{N\sigma_\epsilon^2+\sigma_w^2}$.

Our two theorems together imply that, with an appropriately selected $\lambda \in [0, 1]$, $(1 - \lambda)\mathbf{r}^{\mathrm{OLS}} + \lambda\mathbf{r}^{\mathrm{EO}} = \hat{\mathbf{r}}$. This suggests that directed regression, which optimizes λ via cross-validation to generate a coefficient vector $\mathbf{r}^{\mathrm{DR}} = (1 - \lambda)\mathbf{r}^{\mathrm{OLS}} + \lambda\mathbf{r}^{\mathrm{EO}}$, should approximate $\hat{\mathbf{r}}$ well without observing the missing features or requiring knowledge of Q, σ_ϵ, or σ_w.

2.5.3 Interpretation

To develop intuition for our results, we consider an idealized situation where the coefficients $\mathbf{r}^*$ and $\mathbf{r}^{\perp *}$ are provided to us by an oracle. Then the optimal coefficient vector would be

$$\mathbf{r}^{\mathrm{O}} = \underset{\mathbf{r} \in \mathbb{R}^K}{\operatorname{argmin}} \, \mathrm{E}[\ell(\mathbf{u}_{\mathbf{r}}(\tilde{\mathbf{X}}), \tilde{\mathbf{y}}) | \overline{\mathcal{O}}, \mathbf{r}^*, \mathbf{r}^{\perp *}].$$

It can be shown that $\mathbf{r}^{\mathrm{OLS}}$ is a biased estimator of $\mathbf{r}^{\mathrm{O}}$, while $\mathbf{r}^{\mathrm{EO}}$ is an unbiased one. However, the variance of $\mathbf{r}^{\mathrm{OLS}}$ is smaller than that of $\mathbf{r}^{\mathrm{EO}}$. The optimal tradeoff is indeed captured by the value of λ provided in Theorem 2.2. In particular, as the number of training samples N increases, variance diminishes and λ approaches 1, placing increasing weight on EO. On the other hand, as the number of observed features K increases, model bias decreases and λ approaches 0, placing increasing weight on OLS. Our experimental results demonstrate that the value of λ selected by cross-validation exhibits the same behavior.

2.6 Extensions

One might argue that feature mis-specification is not a critical issue in light of effective methods for subset selection. In particular, rather than selecting a few features and facing the consequences of model bias, one might select an enormous set of features and apply a method like the lasso (Tibshirani, 1996) to identify a small subset. Our view is that even this enormous set will result in model biases that might be ameliorated by generalizations of DR. There is also the concern that data requirements grow with the size of the large feature set, albeit slowly. Understanding how to synthesize DR with subset selection methods can be an interesting direction for

future research.

Another issue that should be explored is the effectiveness of cross-validation in optimizing λ. In particular, it would be helpful to understand how the estimate relates to the ideal value of λ identified by Theorem 2.2. More general work on the selection of convex combinations of models (e.g., Audibert (2004); Bunea et al. (2007)) may lend insights to our setting.

Differences between OLS and EO are analogous to differences that have been studied between generative and discriminative methods for learning (see, e.g., Ng and Jordan (2001)). When data samples are scarce, generative methods often provide better results, as does OLS. On the other hand, when there is ample data, discriminative methods are advantageous, as is the case for EO. DR provides a useful way of combining the merits of OLS and EO, and the idea may generalize to offer an approach that can more broadly be used to combine merits of generative and discriminative methods.

Chapter 3

Directed Time-Series Regression for Control

In this chapter, we extend the application of directed regression from a single-stage decision problem to a multi-stage stochastic control problem. We propose *directed time-series regression*, a new approach to estimating parameters of time-series models for use in certainty equivalent model predictive control. The approach combines merits of least squares regression and empirical optimization. Through a computational study involving a stochastic version of a well known inverted pendulum balancing problem, we demonstrate that directed time series regression can generate significant improvements in controller performance over conventional methods.

3.1 Overview

A common approach to stochastic control, sometimes referred to as *certainty equivalent model predictive control*, involves at each time forecasting future outcomes of exogenous random variables and optimizing control actions over a planning horizon under the assumption that these forecasts will be realized (Bertsekas, 2005; Box et al., 2008). The first of the sequence of actions is taken, and the forecasting and optimization are repeated at each subsequent time step. We propose in this chapter a regression algorithm that fits time series forecasting models for use in such a context.

We focus attention on linear systems with quadratic cost, though the issues and methods we introduce generalize. In particular, we will consider a dynamic system that evolves over discrete time steps $t = 0, 1, 2, \ldots$. At each time t, the state $\mathbf{x}_t \in \mathbb{R}^P$ is updated according to

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{C}\mathbf{w}_t,$$

where $\mathbf{u}_t \in \mathbb{R}^Q$ is a control action, $\mathbf{w}_t \in \mathbb{R}^S$ is a random disturbance, and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are known matrices of appropriate dimension. Each control action u_t is selected just before the disturbance $\mathbf{w}_t$ is observed. The objective is to minimize average expected cost

$$\lim_{h\to\infty} \mathrm{E}\left[\frac{1}{h}\sum_{t=0}^{h-1} g(\mathbf{x}_t, \mathbf{u}_t)\right],$$

where the per period cost function is a positive definite quadratic of the form $g(\mathbf{x}, \mathbf{u}) = \mathbf{x}^{\mathrm{T}}\mathbf{G}_1 x + \mathbf{x}^{\mathrm{T}}\mathbf{G}_2 u + \mathbf{u}^{\mathrm{T}}\mathbf{G}_3 u$, for some $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$. The following example illustrates a specific context.

Example 3.1. *Consider a problem of balancing an inverted pendulum on a cart, as illustrated in Figure 3.1. Take the state to be a four-dimensional vector* $\mathbf{x}_t = [s_t \;\; \theta_t \;\; \dot{s}_t \;\; \dot{\theta}_t]^{\mathrm{T}}$*, where* s_t *is the position of the cart,* θ_t *is the angle of the pendulum, and* $\dot{s}_t$ *and* $\dot{\theta}_t$ *are their rates of change. The cart can move freely along the horizontal axis, and is controlled through applying a voltage* u_t *to its motor. We discretize time, and consider a linearization of the system dynamics around the balance point as in Landry et al. (2005), which in the absence of disturbances can be written as* $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}u_t$ *for some* $\mathbf{A}$ *and* $\mathbf{B}$*. We introduce to the system a disturbance* w_t *which can be thought of as a force exerted by a gust of wind at time* t*. With the disturbances, the linearized system equation becomes* $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}u_t + \mathbf{C}w_t$ *for some* $\mathbf{C}$*. The objective is to center the cart and balance the pole while minimizing energy expenditure, and this is represented by a per period cost function* $g(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^{\mathrm{T}}\mathbf{G}_1\mathbf{x}_t + G_3 u_t^2$*, for some* $\mathbf{G}_1$ *and* G_3*, where the first term captures how far the current state is from being centered and balanced and the second term reflects the energy applied by the control action.*

This example may seem simple and specialized, but our framework captures a broad set of applications. For example, we could model the control of a robot that

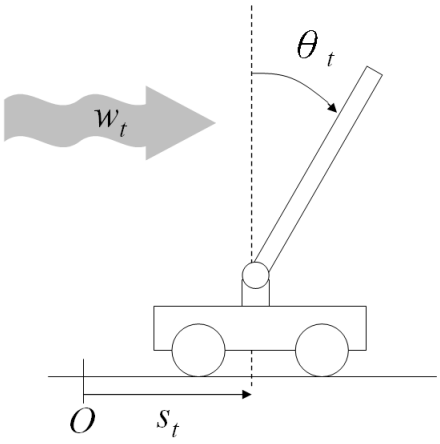


Figure 3.1: Inverted pendulum on a cart.

receives instructions from a human user by encoding in $\mathbf{x}_t$ both the physical state and a tentative plan. Then, $\mathbf{w}_t$ could represent changes to the plan communicated by the user and $\mathbf{u}_t$ could represent an action taken by the robot as it tries to follow the plan.

An important issue is how future disturbances are forecasted. In the inverted pendulum problem, for example, accurate predictions of future wind patterns facilitate more effective control. We consider the use of linear time series models with coefficients estimated based on a historical sequence of disturbances. The main issue we focus on here is how one should estimate coefficients. Note that the setting of linear systems with quadratic cost and linear time series model is one of the most commonly used in applied work and also offers a simple starting point for exploring how learning and control should be coordinated.

As discussed in the previous chapter, it is common to estimate the coefficients of a linear model by minimizing squared error over historical data and then use the model with the resulting coefficients to generate forecasts for model predictive control. However, as we will discuss further, ordinary least squares regression leaves room for improvement when the observed process is not itself generated by the specified time series model. In particular, in such situations it can be beneficial to take the

Notation	Definition
$\mathbf{x}_t$	State vector at time t
$\mathbf{A}$	Dynamics matrix
$\mathbf{B}$	Input matrix of control
$\mathbf{C}$	Input matrix of disturbance
$\mathbf{u}_t/u_t$	Control vector/variable at time t
$\mathbf{w}_t/w_t$	Disturbance vector/variable at time t
$\hat{\mathbf{w}}_{t,\tau}/\hat{w}_{t,\tau}$	The disturbance forecast of time τ made at time t
$g(\mathbf{x}, \mathbf{u})$	Control cost function
M	Length of control horizon
$\mathbf{r}$	Model coefficients vector
$v_{k,t}$	The k-th feature evaluated at time t
$\mu_{\mathbf{r}}$	Policy function parametrized by $\mathbf{r}$
$F_{\mathbf{r}}$	Forecast function parametrized by $\mathbf{r}$
N	Number of observed disturbance samples

Table 3.1: Notation for Chapter 3.

control objective into account when computing coefficients. One approach that does this is empirical optimization, sometimes referred to as empirical risk minimization, which computes coefficients that would have minimized historical average cost given the historical sequence of disturbances. Under certain technical assumptions, if an infinite history of observations is available, empirical optimization yields coefficients that minimize future average expected cost. However, with finite data, empirical optimization often works poorly because it overspecializes to the data.

In this chapter, we propose a new approach to fitting coefficients which we refer to as *directed time-series regression*. This approach combines merits of ordinary least squares regression and empirical optimization. As we will demonstrate through a computational study of the inverted pendulum balancing problem, using directed time series regression can lead to large improvements in controller performance over either of the aforementioned alternatives.

Table 3.1 summarizes our mathematical notation. Part of this chapter has been published in our technical report (Kao and Van Roy, 2010).

3.2 Certainty Equivalent Model Predictive Control

Recall that the problem at hand involves controlling a system that evolves according to

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{C}\mathbf{w}_t,$$

with a goal of minimizing average expected cost, where the per period cost function is a positive definite quadratic of the form $g(\mathbf{x}, \mathbf{u}) = \mathbf{x}^{\mathrm{T}}\mathbf{G}_1\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathbf{G}_2\mathbf{u} + \mathbf{u}^{\mathrm{T}}\mathbf{G}_3 u$.

Although the time horizon of interest is infinite, to simplify the problem, at each time t, model predictive control (MPC) aims to optimize a finite horizon objective of the form

$$\mathrm{E}_t\left[\sum_{\tau=t}^{\tau=t+M} g(\mathbf{x}_\tau, \mathbf{u}_\tau)\right],$$

where M is the horizon time and the subscript t indicates that the expectation is conditioned on $\mathbf{x}_t, \mathbf{w}_{t-1}, \mathbf{w}_{t-2}, \mathbf{w}_{t-3}, \ldots$. After deriving a control policy that optimizes this objective, MPC applies the policy at time t to generate a control action $\mathbf{u}_t$. Then, after observing $\mathbf{w}_t$ and $\mathbf{x}_{t+1}$, a new finite horizon problem is formulated with a horizon spanning times $t+1$ through $t+1+M$. This new problem is solved to produce a control action $\mathbf{u}_{t+1}$. The process is repeated at each subsequent time step.

In order to compute expected future costs, MPC requires a model that provides a distribution over future disturbance sequences. Certainty equivalent MPC simplifies the problem by assuming that future disturbances follow a deterministic sequence $\hat{\mathbf{w}}_{t,t}, \hat{\mathbf{w}}_{t,t+1}, \hat{\mathbf{w}}_{t,t+2}, \ldots$ generated at time t by a forecasting model. Hence, the objective of the optimization problem solved at time t becomes

$$\sum_{\tau=t}^{\tau=t+M} g(\mathbf{x}_\tau, \mathbf{u}_\tau), \tag{3.1}$$

where the state dynamics are governed by the same linear difference equation as before but with disturbances given by $(\mathbf{w}_t, \mathbf{w}_{t+1}, \mathbf{w}_{t+2}, \ldots) = (\hat{\mathbf{w}}_{t,t}, \hat{\mathbf{w}}_{t,t+1}, \hat{\mathbf{w}}_{t,t+2}, \ldots)$.

For our context, which involves a linear system and a quadratic cost function, certainty equivalent MPC generates a control policy that is linear in state and forecasts. To simplify our discussion we will assume that disturbances are scalar-valued, though the ideas and methods we will present extend to the vector-valued case. Under this assumption, there are matrices $\mathbf{L}$ and $\mathbf{H}$, which depend only on $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{G}_1$, $\mathbf{G}_2$, G_3, and M, such that at each time t, the action employed by certainty equivalent MPC is given by

$$\mathbf{u}_t = \mathbf{L}\mathbf{x}_t + \mathbf{H}\hat{w}_{t,t}^{t,t+M-1}, \tag{3.2}$$

where $\hat{w}_{t,t}^{t,t+M-1} = [\hat{w}_{t,t} \ \cdots \ \hat{w}_{t,t+M-1}]^{\mathrm{T}}$. In particular, this action minimizes the objective in (3.1).

Certainty equivalent MPC leads to effective decisions for a wide variety of control problems. For example, in our context of linear-quadratic control, if the process that generates disturbances is known and ergodic and each forecast $\hat{w}_{t,t+\tau}$ is equal to the expected disturbance $\mathrm{E}_t[w_{t+\tau}]$ then the performance of certainty equivalent MPC becomes arbitrarily close to optimal as M grows. Certainty equivalent MPC has also been applied with success to a number of important nonlinear systems.

3.3 Forecasting Model

We consider the use of a model driven by K linear features

$$v_{k,t} = \sum_{\tau=1}^{T} \phi_{k,\tau} w_{t-\tau}, \qquad k = 1, \ldots, K.$$

Here, each $\phi_{k,\tau}$ is a scalar constant and each $v_{k,t}$ represents a scalar feature that is useful for predicting the next disturbance. We consider a model of the form

$$w_t = \sum_{k=1}^{K} \mathbf{r}_k v_{k,t} + z_t,$$

where $\mathbf{r} \in \mathbb{R}^K$ is a vector of model coefficients and $z_t \sim \mathcal{N}(0, \sigma_z^2)$ is i.i.d Gaussian noise. Note that the conditional expectation of disturbance w_t at time t is linear in

past disturbances.

It is straightforward to generate forecasts $\hat{w}_{t,t}, \hat{w}_{t,t+1}, \ldots \hat{w}_{t,t+M-1}$ given the coefficients $\mathbf{r}$, features $\phi_1, \ldots, \phi_K$, and past disturbances $w_{t-1}, \ldots, w_{t-T}$. In particular, let $\hat{w}_{t,\ell} = w_\ell$ for $\ell < t$, and for $\ell = t, \ldots, t+M-1$, recursively compute

$$\hat{v}_{k,t,\ell} = \sum_{\tau=1}^{T} \phi_{k,\tau} \hat{w}_{t,\ell-\tau}, \qquad k = 1, \ldots, K,$$

and

$$\hat{w}_{t,\ell} = \sum_{k=1}^{K} \mathbf{r}_k \hat{v}_{k,t,\ell}.$$

In the event that the data is generated by our model, each forecast produced in this way is a conditional expectation: $\hat{w}_{t,t+\tau} = \mathrm{E}_t[w_{t+\tau}]$.

Since, conditioned on $w_{t-1}, \ldots, w_{t-T}$, the forecasts $\hat{w}_{t,t}, \ldots, \hat{w}_{t,t+M-1}$ are conditionally independent of previous disturbances, there is a function $F_{\mathbf{r}}$ which maps the T most recent disturbances to forecasts of the next M disturbances. In particular,

$$\hat{w}_{t,t}^{t,t+M-1} = F_{\mathbf{r}}(w_{t-T}^{t-1}),$$

where $w_{t-T}^{t-1} = [w_{t-T} \ \cdots \ w_{t-1}]^{\mathrm{T}}$. Note that $F_{\mathbf{r}}$ depends on the coefficient vector $\mathbf{r}$ and is generally not linear in $\mathbf{r}$. From (3.2) we see that control actions generated by certainty equivalent MPC based on forecasts produced by our model can be written as

$$\mathbf{u}_t = \mathbf{L}\mathbf{x}_t + \mathbf{H}F_{\mathbf{r}}(w_{t-T}^{t-1}).$$

Let $\mu_{\mathbf{r}}$ denote a control policy that maps $\mathbf{x}_t$ and w_{t-T}^{t-1} to $\mathbf{u}_t$. Note that this control policy depends on $\mathbf{r}$ and again is generally not linear in $\mathbf{r}$.

3.4 Time-Series Regression Algorithms

We now discuss algorithms for computing coefficients to fit a forecasting model of the kind we have described to a time series. In particular, each algorithm will compute a

vector $\mathbf{r} \in \mathbb{R}^K$ given observed samples $w_0, \dots w_{N-1}$.

3.4.1 Ordinary Least Squares

As discussed in the previous chapter, the most common approach is ordinary least squares regression (OLS). Here, we compute coefficients

$$\mathbf{r}^{\text{OLS}} := \operatorname*{argmin}_{\mathbf{r}} \sum_{t=T}^{N-1} \left\| w_t - \sum_{k=1}^{K} \mathbf{r}_k v_{k,t} \right\|_2^2 .$$

Note that the sum begins with $t = T$ because earlier samples $w_0, \dots, w_{T-1}$ are required to compute the features $v_{k,T}$. The optimization problem can be solved efficiently since it is a linear least squares problem. The resulting coefficient vector $\mathbf{r}^{\text{OLS}}$ represents the maximum likelihood estimate assuming that the observed time series is generated by our forecasting model. As such, this is a natural approach to fitting a forecasting model that accurately captures the generating process.

3.4.2 Empirical Optimization

OLS does not take the control problem into account when computing coefficients. This assumes a separation principle whereby model fitting can be decoupled from the subsequent selection of control actions. Empirical optimization (EO) provides an alternative that does account for the control problem when computing coefficients. The algorithm computes coefficients that minimize "historical cost":

$$\mathbf{r}^{\text{EO}} := \operatorname*{argmin}_{\mathbf{r}} \sum_{t=T}^{N-1} g(\mathbf{x}_t^{\mu_\mathbf{r}}, \mu_\mathbf{r}(\mathbf{x}_t^{\mu_\mathbf{r}}, w_{t-T}^{t-1})), \tag{3.3}$$

where $\mathbf{x}_T^{\mu_\mathbf{r}}$ is set to the initial state of the system and subsequent control actions $\mathbf{u}_T, \mathbf{u}_{T+1}, \dots, \mathbf{u}_{N-1}$ are selected by $\mu_\mathbf{r}$. The objective here is the sum of costs that would have been realized if we were to apply certainty equivalent MPC alongside our forecast model with coefficients $\mathbf{r}$, starting at time T. It is easy to see that if the disturbance process is ergodic, as the number N of observed samples increases,

historical average cost converges to future average cost, and therefore, EO computes coefficients that optimize future average cost. However, for reasonable values of N, EO tends to overspecialize to the data, and this results in poor future performance.

It may appear difficult to compute $\mathbf{r}^{\mathrm{EO}}$ because $\mu_{\mathbf{r}}$ is generally not linear in $\mathbf{r}$. However, as in the case of the computational study we will later discuss, we have had positive experience applying local optimization methods initialized with $\mathbf{r}^{\mathrm{OLS}}$. A more detailed description of our implementation is given in the appendix.

Nevertheless, this non-convex optimization problem can still be challenging as the number of parameters K increases, mainly because evaluating the gradients and Hessians of $F_{\mathbf{r}}$ requires iterative computation. We now propose an approximation algorithm that relieve this problem by linearization. Note that $\mu_{\mathbf{r}}$ is typically close to linear in a region of $\mathbb{R}^K$ that includes reasonable choices of $\mathbf{r}$, for reasons we now explain. The policy $\mu_{\mathbf{r}}$ can be written as a sum of terms that are linear and terms that are nonlinear in $\mathbf{r}$. Linear terms are multiples of terms of the form $\mathbf{r}_k\phi_{k,\tau}$. Nonlinear terms are higher order terms that involve products of terms of the form $\mathbf{r}_k\phi_{k,\tau}$. In most problems of practical interest, terms of the form $\mathbf{r}_k\phi_{k,\tau}$ are significantly smaller than one because past disturbances exhibit decaying influence on future ones. As such, the nonlinear terms tend to be significantly smaller than the linear ones, which therefore play a dominant role in $\mu_{\mathbf{r}}$. This motivates the following approximation algorithm.

Suppose we have a reasonable estimate $\hat{\mathbf{r}}$ for the model coefficients, which we will refer to as base coefficients. In practice, a typical choice for it is $\mathbf{r}^{\mathrm{OLS}}$. Given $\hat{\mathbf{r}}$, it is straightforward to iteratively generate forecasts $\hat{w}_{t,t}, \hat{w}_{t,t+1}, \ldots, \hat{w}_{t,t+M-1}$ and features $\hat{v}_{k,t,t}, \hat{v}_{k,t,t+1}, \ldots, \hat{v}_{k,t,t+M-1}$, $k = 1, \cdots, K$. Now fix these features and define

$$\tilde{w}_{t,\ell} := \sum_{k=1}^{K} \mathbf{r}_k \hat{v}_{k,t,\ell}, \quad \ell = t, \ldots, t+M-1.$$

In other words, here we fix base coefficients $\hat{\mathbf{r}}$ when rolling forward but allow another set of coefficients $\mathbf{r}$ to take effect at the last step of prediction. Apparently $\tilde{w}_{t,\ell}$ is

linear in $\mathbf{r}$. Furthermore, by similar notation we define

$$\tilde{F}_{\hat{\mathbf{r}},\mathbf{r}}(w_{t-T}^{t-1}) := \tilde{w}_{t,t}^{t,t+M-1}$$

and

$$\tilde{\mu}_{\hat{\mathbf{r}},\mathbf{r}}(\mathbf{x}_t, w_{t-T}^{t-1}) := \mathbf{L}\mathbf{x}_t + \mathbf{H}\tilde{F}_{\hat{\mathbf{r}},\mathbf{r}}(w_{t-T}^{t-1}),$$

both of which are also (affine) linear in $\mathbf{r}$. As we can see, this approximation trades in some model flexibility for linearity. This leads to an alternative formulation for EO, which we refer to as linearized EO (LEO):

$$\mathbf{r}^{\text{LEO}} := \operatorname*{argmin}_{\mathbf{r}} \sum_{t=T}^{N-1} g(\mathbf{x}_t^{\tilde{\mu}_{\hat{\mathbf{r}},\mathbf{r}}}, \tilde{\mu}_{\hat{\mathbf{r}},\mathbf{r}}(\mathbf{x}_t^{\tilde{\mu}_{\hat{\mathbf{r}},\mathbf{r}}}, w_{t-T}^{t-1})). \tag{3.4}$$

Unlike (3.3), (3.4) can be solved efficiently by quadratic programming. A policy function $\tilde{\mu}_{\hat{\mathbf{r}},\mathbf{r}^{\text{LEO}}}$ can then be built upon $\mathbf{r}^{\text{LEO}}$.

3.4.3 Directed Time-Series Regression

Directed time-series regression (DTSR) aims to combine the merits of OLS and EO. A naive version of DTSR (NDR) produces a vector of coefficients that is a convex combination of those computed by OLS and EO:

$$\mathbf{r}^{\text{NDR}} := (1-\lambda)\mathbf{r}^{\text{OLS}} + \lambda\mathbf{r}^{\text{EO}},$$

where $\lambda \in [0,1]$ is selected by cross-validation. This algorithm, albeit simple and intuitive, requires solving EO as an intermediate step and therefore can be inefficient for complicated problems. This disadvantage motivates the following approximation algorithm, linearized DTSR (LDR), which produces a coefficient vector by taking a convex combination of $\mathbf{r}^{\text{OLS}}$ and $\mathbf{r}^{\text{LEO}}$

$$\mathbf{r}^{\text{LDR}} := (1-\lambda)\mathbf{r}^{\text{OLS}} + \lambda\mathbf{r}^{\text{LEO}}.$$

A policy function $\tilde{\mu}_{\hat{\mathbf{r}},\mathbf{r}^{\text{LDR}}}$ then follows.

We will use a sliding window cross-validation procedure to select the parameter $\lambda \in [0, 1]$ for NDR and LDR. The windows are defined by a sequence $t_i \in \{T+1, T+2, \ldots, N-1\}$, each element specifying a boundary that separates the ith training set window from the ith validation set window. To select the parameter λ for NDR, for each i we compute $\mathbf{r}^{\mathrm{OLS}}$ and $\mathbf{r}^{\mathrm{EO}}$ using the observations $w_0, w_1, \ldots, w_{t_i}$, and then evaluate each choice of λ_i by simulating the system with control actions generated by $\mu_{(1-\lambda)\mathbf{r}^{\mathrm{OLS}}+\lambda\mathbf{r}^{\mathrm{EO}}}$, starting at state $\mathbf{x}_{t_i} = 0$. The performance of each λ_i is judged based on the sum of costs incurred over time $t_i + 1, \ldots, N-1$. For each i, the best value of λ_i is selected, and we take λ to be the average over i among the selected values of λ_i. In our computational study, we set t_1, t_2, and t_3 to $T + 0.3(N-T), T + 0.5(N-T)$, and $T + 0.7(N-T)$, each rounded off to the closest integer. The cross-validation procedure for LDR is essentially the same, except the replacement of $\mathbf{r}^{\mathrm{EO}}$ by $\mathbf{r}^{\mathrm{LEO}}$ and $\mu_{(1-\lambda)\mathbf{r}^{\mathrm{OLS}}+\lambda\mathbf{r}^{\mathrm{EO}}}$ by $\tilde{\mu}_{\hat{\mathbf{r}},(1-\lambda)\mathbf{r}^{\mathrm{OLS}}+\lambda\mathbf{r}^{\mathrm{LEO}}}$. Figure 3.2 illustrates this sliding window cross-validation procedure.

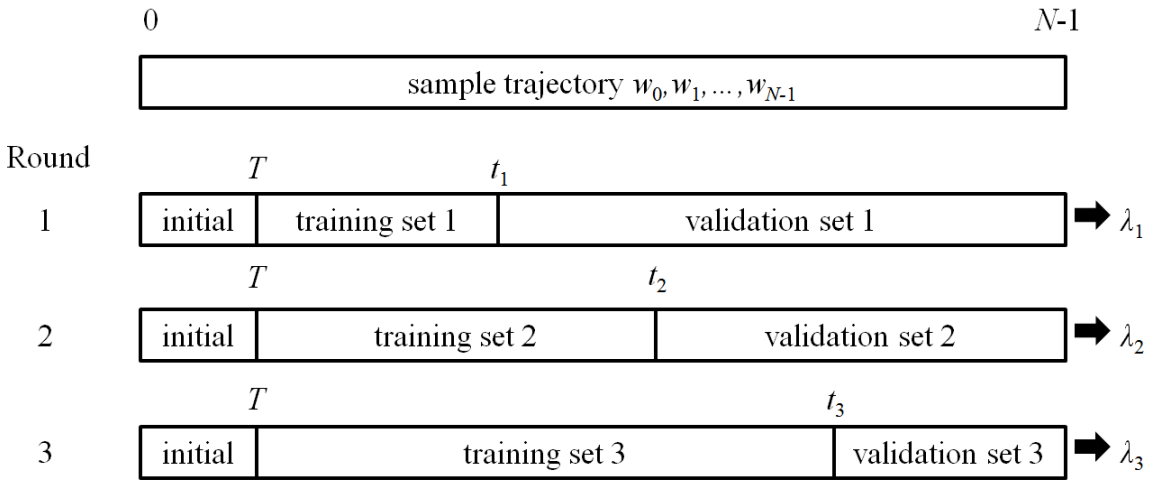


Figure 3.2: Sliding window cross-validation.

3.5 Computational Study

To assess relative merits of the algorithms we have discussed, we conducted a computational study involving an inverted pendulum problem of the kind described in Example 3.1.

3.5.1 System Dynamics

We discretize time, sampling at a rate of 100Hz. We employ a linearization of this system around the balance point, as derived in Landry et al. (2005). We also use the same physical parameters therein, and we assume that at each time t there is a gust of wind accelerating the pendulum's angle by w_t. Specifically, the system dynamics are characterized by the following matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0.01 & 0 \\ 0 & 1 & 0 & 0.01 \\ 0 & -0.0178 & 0.8872 & 0 \\ 0 & 0.2847 & 0.2773 & 1 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0.0198 \\ -0.04871 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.01 \end{bmatrix}.$$

Since our goal is to keep the cart centered and the pendulum vertical while minimizing energy consumption, we use a cost function parameterized by the following matrices:

$$\mathbf{G}_1 = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, G_2 = 0, G_3 = 0.1.$$

Hence, the cost in period t is $\mathbf{x}_t^{\mathrm{T}}\mathbf{G}_1\mathbf{x}_t + 0.1u_t^2$, where the state is $\mathbf{x}_t = [s_t \;\; \theta_t \;\; \dot{s}_t \;\; \dot{\theta}_t]^{\mathrm{T}}$ and the action u_t influences acceleration. Along with the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{G}_1, G_2$, and G_3, we take M to be 100 in our experiment and evaluate the policy matrices $\mathbf{L}$ and $\mathbf{H}$ accordingly.

3.5.2 Generative and Forecasting Models

In our study, we randomly sample an ensemble of generative models, each of which is used to generate a time series to which our algorithms are applied. Each generative model is sampled as follows:

1. Sample $\psi_1, \psi_2, \ldots, \psi_5$ i.i.d from $\mathcal{N}(0, 1)$.
2. Sample $\psi_6, \psi_7, \ldots, \psi_{30}$ i.i.d from $\mathcal{N}(0, \frac{1}{10^2})$.
3. Select $\beta \in \mathbb{R}_+$ such that the process

$$w_t = \beta \sum_{\tau=1}^{30} \psi_\tau w_{t-\tau} + z_t$$

   has a variance $\mathrm{E}[w_t^2] = 2\sigma_z^2$.
4. Select σ_z^2 such that when control actions are selected according to $u_t = \mathbf{L}\mathbf{x}_t$, the average expected cost is equal to one.

These steps result in an AR(30) process for which the first five coefficients tend to be an order of magnitude larger than the next twenty-five. The last two steps in the sampling process deserve some explanation. Step 3 normalizes the signal-to-noise ratio of the disturbance process. This reduces variations among sampled generative models in how helpful a forecasting model can be. Step 4 serves to normalizes the control objective which would otherwise dramatically differ from one generative model to another. The control policy $u_t = \mathbf{L}\mathbf{x}_t$ is one that is optimal when future disturbances are not forecastable. The disturbance variance σ_z^2 is chosen so that the average expected cost when the system is controlled by this naive policy is equal to one.

Since the five most recent disturbances dominate others in influencing future disturbances, it is natural to select them as features of a forecasting model. In particular,

we consider a forecasting model with five features:

$$v_{k,t} = w_{t-k}, \qquad k = 1, \ldots, 5.$$

This model approximates the generative model but at the same time neglects useful information that can be extracted from disturbances observed beyond five time periods in the past. It is our intention to consider a forecasting model that does not perfectly capture the generative model since it is in such contexts that DTSR adds value. It is also important to recognize that this sort of model misspecification is inevitable in practical applications.

3.5.3 Results

For our computational study, we sampled 2,000 generative models using the above procedure. For each model, a sequence of 5,360 disturbances $w'_0, \ldots, w'_{5359}$ are generated, initialized with an assumption that all disturbances prior to w'_0 are equal to zero. We take the last N samples of this trajectory as observations to be used by regression algorithms. In other words, each algorithm makes use of disturbances $w_t = w'_{5360-N+t}$ for $t = 0, 1, \ldots, N-1$. In our experiment, we repeat the OLS, EO, NDR, LEO, and LDR algorithms for each $N \in \{200, 240, \ldots, 360\}$.

To provide a lower bound, we consider a "model-clairvoyant (MC) policy." This is the optimal policy to use given full knowledge of the generative model. The average expected cost g^* incurred by this policy, which we will refer to as the MC cost, can be derived in closed form. Clearly, policies generated by the aforementioned five algorithms must incur average expected costs no less than g^*.

We can also evaluate in closed form the average expected costs g^{OLS}, g^{EO}, g^{NDR}, g^{LEO}, and g^{LDR}, of policies generated by our regression algorithms. We will focus our presentation on excess costs $g^{\mathrm{OLS}} - g^*$, $g^{\mathrm{EO}} - g^*$, $g^{\mathrm{NDR}} - g^*$, $g^{\mathrm{LEO}} - g^*$, and $g^{\mathrm{LDR}} - g^*$. Subtracting g^* factors out the influence of costs that are inevitable regardless of the choice of control policy. Figure 3.3 plots excess costs as a function of N. These results are qualitatively similar to those reported in the previous chapter in a context involving repeated independent decision problems rather than intertemporal control.

For small N, EO and LEO suffer from overspecialization. For large N, the degree of overspecialization diminishes and the bias in OLS due to model mispecification prevents OLS from doing as well as EO and LEO. LDR always outperforms the best of these three methods, sometimes by as much as 12%. It is also interesting to note that LDR provides a consistent gain over NDR, in spite of the substantial saving in computation.

We can also compare quantities with simple physical interpretations. Suppose this cart-pendulum system is considered to reach a state of failure if the cart's position or the pendulum's angle deviates from zero by more than certain threshold values. Let the thresholds on position and angle be $B_s = 0.0392$ and $B_\theta = 0.0364$, which are roughly twice of the root-mean-square values of s_t and θ_t in our simulations. We refer to the number of time periods elapsed before a system initialized at the zero state reaches failure as *time-until-failure*. For each of our 2,000 models, we generate 50 disturbance trajectories. For each of these trajectories and each of the policies under consideration (MC, OLS, EO, NDR, LEO, and LDR), we simulated the system until failure using a training data set of size 240. Figure 3.4 summarizes results. It is clear that NDR and LDR are much closer to MC than OLS, EO, and LEO are in terms of time-until-failure.

3.6 Summary

In this chapter we have presented directed time-series regression, an algorithm that computes coefficients of a time-series model for use in certainty equivalent model predictive control. Two versions of this algorithm have been proposed, namely NDR and LDR, which we will collectively refer to as DTSR in the following discussion. We have shown that when the quantity of available data is limited and the forecasting model differs from the generative model, this algorithm offers a significant advantage by combining merits of least squares regression and empirical optimization. As side products, we have also proposed a methodology that transforms the original problem into a linearized version, and a sliding window cross-validation algorithm for time series with control. Both techniques can potentially enlarge the applicability of directed

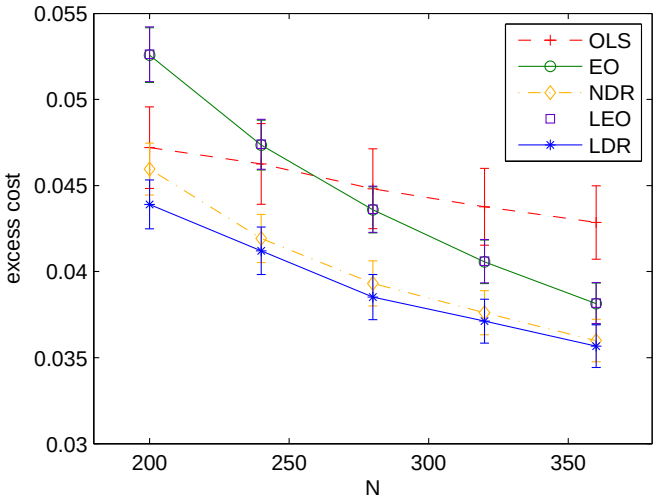


Figure 3.3: The excess costs delivered by OLS, EO, NDR, LEO, and LDR for different numbers of training samples N.

regression theory in other fields.

The main point here is that the coordination of forecasting and control can be fruitful and we hope this observation will stimulate work along these lines involving broader classes of control problems and learning algorithms. Our approach extends to more complex settings than the one considered in this chapter, for example, cases where disturbances are multidimensional and the system is nonlinear. It would also be interesting to explore the use of DTSR in contexts involving non-stationary time series.

Let us summarize with a discussion of several threads of work that relate to the ideas we have presented. Motivated by a similar perspective on fitting a mispecified model for use by a control algorithm, the approach developed in Abbeel and Ng (2004) learns a first-order Markov model in a way that improves controller performance when data is not generated by a first-order Markov model. This approach takes the discount factor of the control problem into account when learning the transition matrix. Aside from the contextual differences between discrete Markov processes and linear autoregressive models, a conceptual advance in our work can be seen in the

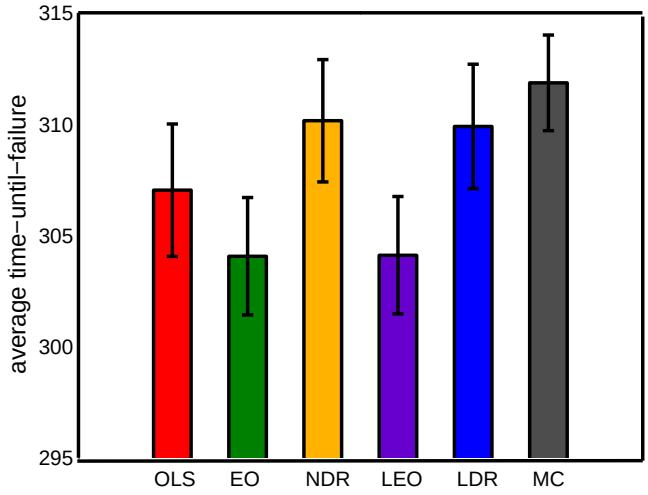


Figure 3.4: The average time-until-failure delivered by MC, OLS, EO, NDR, LEO, and LDR.

fact that DTSR takes the entire control objective into account.

Control theorists working on system identification typically adopt weighted least-square linear regression to fit a linear system (Ljung, 1998). While this approach puts more emphasis on learning the critical components, it does not explicitly consider the control objective. Econometricians have analyzed properties of parameter estimates for misspecified time series models (White, 1980; Domowitz and White, 1982). However, this line of work does not treat the use of such models and estimated parameters for decision or control. Operations researchers have developed methods that factor objectives into how point estimates such as forecasts are generated when they are to be used for decision or control (Granger, 1969; Liyanage and Shanthikumar, 2005; Chua et al., 2008). However, this line of work does not consider model misspecification.

Chapter 4

Directed Principal Component Analysis

In this chapter, we consider a problem involving estimation of a high-dimensional covariance matrix that is the sum of a diagonal matrix and a low-rank matrix, and making a decision based on the resulting estimate. Such problems arise, for example, in portfolio management, where a common approach employs principal component analysis (PCA) to estimate factors used in constructing the low-rank term of the covariance matrix. The decision problem is typically treated separately, with the estimated covariance matrix taken to be an input to an optimization problem. We propose *directed PCA*, an efficient algorithm that takes the decision objective into account when estimating the covariance matrix. Directed PCA effectively adjusts factors that would be produced by PCA so that they better guide the specific decision at hand. We demonstrate through computational studies that directed PCA yields significant benefit, and we prove theoretical results establishing that the degree of improvement over conventional PCA can be unbounded.

4.1 Overview

We consider a problem that involves estimating a covariance matrix from independent identically distributed sample vectors and then making a decision based on this

estimate. The payoff depends on the decision and an additional data sample, observed after the decision is made and generated independently according to the same distribution as those previously observed. We focus on the case where the dimension of sample vectors is large relative to the number of observed samples.

Our formulation is relevant, for example, to the area of portfolio management, where it is common to estimate asset return covariances and to use these estimates to guide investment decisions. The prototypical decision problem here is to select a portfolio that maximizes risk-adjusted return, with risk measured in terms of return variance, and this optimization problem takes the form of a quadratic program in which return expectations and covariances serve as problem data.

To produce a meaningful estimate of a high-dimensional covariance matrix from a limited number of samples, we must assume that the matrix obeys some simplifying structure. In this chapter we consider a scenario where the covariance matrix can be well-approximated by the sum of a diagonal matrix and a low-rank symmetric matrix. Such a covariance matrix can be viewed as representing a factor model, in which each observed variable is a noise-corrupted linear combination of latent common factors. A common approach to estimating such a covariance matrix is through principal component analysis (PCA). This approach focusses on explaining the observed data without regard to the objective of the subsequent decision. In other words, covariance matrix estimation and decision optimization are carried out independently. This separation leaves much room for improvement, as we shall demonstrate in this chapter.

We propose a new approach – *directed PCA* – which estimates the covariance matrix in a way that is tailored to the objective of the subsequent decision problem. As with PCA, directed PCA produces an estimate that is the sum of a diagonal matrix and a low-rank symmetric matrix. It essentially carries out empirical optimization, but subject to a constraint that the resulting covariance matrix explains the data well. In particular, the point estimate is selected from a confidence region around a MAP estimate. Thus, directed PCA aims to combine the merits of empirical optimization and PCA-based methods for covariance estimation.

A new formulation for estimating covariance matrices from high-dimensional data

is not practically useful without an efficient estimation algorithm. An important contribution of this work is an efficient algorithm for directed PCA, with computational requirements comparable to those of conventional PCA. To assess merits of directed PCA, we apply the algorithm to study two data sets: one is a synthetic data set designed for this specific purpose, while the other is an empirical time series of S&P 500 stock returns. We find that directed PCA yields significant improvements over conventional approaches. Specifically, applying directed PCA to a portfolio management example based on empirical data increases certain-equivalent payoff by 7%. With our synthetic data set, we identify plausible examples where the gain reaches 34%.

Aside from our formulation, algorithm, and empirical study, a significant contribution of this work is in a pair of theoretical results that elucidate the benefits of directed PCA. These results assume that the covariance matrix of the generating distribution is the sum of a diagonal matrix and a low-rank symmetric matrix. As such, each data sample can be viewed as a noise-corrupted linear combination of factor loading vectors. The first of our two theoretical results establishes that when the decision objective is aligned with one of the factor loading vectors, the absolute performance increase from using directed PCA instead of MAP estimation can grow linearly in the dimension of the data. The second result establishes that when the decision objective is orthogonal to the span of factor loading vectors, the percentage performance increase from using directed PCA instead of MAP estimation can grow linearly in the dimension of the data. These two results offer striking indications of the importance of accounting for the decision objective in the estimation process, especially when dealing with high-dimensional data.

Table 4.1 summarizes our mathematical notation.

4.2 Problem Formulation

Consider a stochastic optimization problem that involves selecting a decision $\mathbf{u} \in \mathbb{R}^M$ in order to maximize the expected value of an objective function

$$g(\mathbf{u}, \mathbf{x}) = \mathbf{c}^{\mathrm{T}}\mathbf{u} - (\mathbf{u}^{\mathrm{T}}\mathbf{x})^2, \tag{4.1}$$

Notation	Definition
$\mathbf{x}_n$	The n-th sample vector
M	Dimension of data
$\mathcal{X}$	Dataset
$\mathbf{u}$	Decision vector
$g(\mathbf{u}, \mathbf{x})$	Decision objective
$\mathbf{c}$	Objective vector
$\mathbf{\Sigma}_*$	True covariance matrix
$\mathbf{F}$	Factor loadings matrix
σ^2 / $\mathbf{R}$	Uniform / Nonuniform residual variances
$\mathcal{F}_\lambda$	Soft-thresholding operator
$\mathbf{T}$	Rescaling matrix
$\mathbb{S}_+^M$	All symmetric and positive semidefinite matrices of dimension M

Table 4.1: Notation for Chapter 4.

where $\mathbf{c} \in \mathbb{R}^M$ is a given objective vector, and $\mathbf{x} \in \mathbb{R}^M$ is a random vector drawn from a zero-mean Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma}_*)$, where $\mathbf{\Sigma}_*$ is unknown to us. This problem could be easily solved if we had access to the covariance matrix $\mathbf{\Sigma}_*$. Indeed, since

$$\mathrm{E}[g(\mathbf{u}, \mathbf{x})] = \mathbf{c}^{\mathrm{T}}\mathbf{u} - \mathbf{u}^{\mathrm{T}}\mathbf{\Sigma}_*\mathbf{u},$$

we can rewrite this decision problem as

$$\max_{\mathbf{u} \in \mathbb{R}^M} \mathbf{c}^{\mathrm{T}}\mathbf{u} - \mathbf{u}^{\mathrm{T}}\mathbf{\Sigma}_*\mathbf{u}, \tag{4.2}$$

which has an analytical solution $\mathbf{u}_* = \frac{1}{2}\mathbf{\Sigma}_*^{-1}\mathbf{c}$. Now suppose we have observed N sample vectors drawn i.i.d. from $\mathcal{N}(0, \mathbf{\Sigma}_*)$, denoted by a set $\mathcal{X} = \{\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(N)}\}$. Our goal is to compute a point estimate $\hat{\mathbf{\Sigma}}$ for $\mathbf{\Sigma}_*$ based on the dataset $\mathcal{X}$, and use that estimate to guide our decision making. One natural approach of doing so is to use $\hat{\mathbf{\Sigma}}$ as a surrogate for $\mathbf{\Sigma}_*$ in the decision problem (4.2). This leads to a decision

$$\hat{\mathbf{u}} = \operatorname*{argmax}_{\mathbf{u} \in \mathbb{R}^M} \mathbf{c}^{\mathrm{T}}\mathbf{u} - \mathbf{u}^{\mathrm{T}}\hat{\mathbf{\Sigma}}\mathbf{u} = \frac{1}{2}\hat{\mathbf{\Sigma}}^{-1}\mathbf{c}. \tag{4.3}$$

The out-of-sample performance of this resulting decision, given by $\mathrm{E}[g(\hat{\mathbf{u}}, \mathbf{x})|\boldsymbol{\Sigma}_*] = \mathbf{c}^{\mathrm{T}}\hat{\mathbf{u}} - \hat{\mathbf{u}}^{\mathrm{T}}\boldsymbol{\Sigma}_*\hat{\mathbf{u}}$, is therefore the evaluation criterion for an estimate $\hat{\boldsymbol{\Sigma}}$.

A typical Bayesian treatment of this problem goes as follows. We first choose a prior distribution $p(\boldsymbol{\Sigma})$ that reflects our belief about the properties of $\boldsymbol{\Sigma}_*$. We then evaluate the posterior probability $p(\boldsymbol{\Sigma}|\mathcal{X})$ conditioned on observation $\mathcal{X}$, and solve for a Bayes optimal decision by optimizing

$$\max_{\mathbf{u}} \mathrm{E}[g(\mathbf{u}, \mathbf{x})|\mathcal{X}]. \tag{4.4}$$

Recall that

$$\mathrm{E}[g(\mathbf{u}, \mathbf{x})|\mathcal{X}] = \int_{\boldsymbol{\Sigma}} \mathrm{E}[g(\mathbf{u}, \mathbf{x})|\boldsymbol{\Sigma}]p(\boldsymbol{\Sigma}|\mathcal{X})d\boldsymbol{\Sigma}. \tag{4.5}$$

This integral is generally hard to evaluate, and one common relaxation is to assume $p(\boldsymbol{\Sigma}|\mathcal{X})$ peaks sharply at its mode. Specifically, let $\boldsymbol{\Sigma}_{\mathrm{MAP}}$ denote the *maximum a posteriori* (MAP) estimate, defined as

$$\boldsymbol{\Sigma}_{\mathrm{MAP}} = \operatorname*{argmax}_{\boldsymbol{\Sigma}} p(\boldsymbol{\Sigma}|\mathcal{X}). \tag{4.6}$$

We can approximate (4.5) by

$$\mathrm{E}[g(\mathbf{u}, \mathbf{x})|\mathcal{X}] \simeq \int_{\boldsymbol{\Sigma}} \mathrm{E}[g(\mathbf{u}, \mathbf{x})|\boldsymbol{\Sigma}]\delta(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{\mathrm{MAP}})d\boldsymbol{\Sigma} = \mathrm{E}[g(\mathbf{u}, \mathbf{x})|\boldsymbol{\Sigma}_* = \boldsymbol{\Sigma}_{\mathrm{MAP}}], \tag{4.7}$$

which, together with (4.4) and (4.3), suggest a decision $\mathbf{u} = \frac{1}{2}\boldsymbol{\Sigma}_{\mathrm{MAP}}^{-1}\mathbf{c}$. Thus, instead of solving a convoluted decision problem, practitioners often adopt the above rationale and compute the MAP estimate as a solution.

To produce a meaningful MAP estimate from a limited number of samples in high-dimensional space, the prior $p(\boldsymbol{\Sigma})$ is usually designed to put stronger weights on those covariance matrices that obey some simplifying structure. In this work, we focus on a scenario in which the covariance matrix $\boldsymbol{\Sigma}_*$ is believed to be dominated by a few components. Mathematically speaking, we will assume that $\boldsymbol{\Sigma}_*$ can be well-approximated by the sum of a diagonal matrix $\mathbf{R}_* \succeq 0$ and a symmetric matrix $\mathbf{F}_* \succeq 0$ whose rank is much smaller than the dimension M. This assumption effectively

corresponds to a factor model that generates each sample vector by

$$\mathbf{x}_{(n)} = \mathbf{F}_*^{\frac{1}{2}} \mathbf{z}_{(n)} + \mathbf{w}_{(n)},$$

where $\mathbf{F}_*^{\frac{1}{2}}$ is a thin matrix that represents the factor loadings, $\mathbf{z}_{(n)} \sim \mathcal{N}(0, \mathbf{I})$ represents a parsimonious set of independent factors, and $\mathbf{w}_{(n)} \sim \mathcal{N}(0, \mathbf{R}_*)$ represents the idiosyncratic residual noise not captured by these factors. Such a model has been widely applied in economics, finance, medicine, psychology, and various other natural and social sciences (Harman, 1976). We will discuss two popular choices of prior that express this factor model assumption in next section.

Although using MAP estimate to guide decision making is a common approach widely adopted by practitioners due to its simplicity, it generally suffers from two disadvantages. First, the prior $p(\mathbf{\Sigma})$ is usually chosen in a way that not only reflects our assumption but also enables efficient computation of the MAP estimate. Such mathematical convenience is often attained at the price of introducing model bias, or equivalently, prior mis-specification. Second, MAP estimation does not take into account the decision objective when computing an estimate. As we will see in the following sections, these disadvantages together leave room for improvement.

4.3 Learning Algorithms: Uniform Residual Case

We will begin our discussion with a simplified scenario in which the residual variances are further assumed to be identical. In other words, we will assume $\mathbf{R}_*$ is a multiple of identity matrix, denoted by $\sigma_*^2\mathbf{I}$, and therefore $\mathbf{\Sigma}_* = \mathbf{F}_* + \sigma_*^2\mathbf{I}$. This assumption helps us better illustrate our main idea, and will be relaxed in the next section.

4.3.1 Regularized Maximum-Likelihood Estimates

Instead of solving (4.6) directly for an MAP estimate, in practice one might convert it into a *regularized* maximum-likelihood estimation problem, where the regularization reflects our prior belief that $\mathbf{\Sigma}_*$ obeys the factor model assumption. We now consider two types of regularization for this purpose, both of which are popular partly due to

the fact that they can be efficiently computed via PCA.

Constraining the Rank

Since we assume the data is generated by a factor model with a parsimonious set of factors, a natural choice of regularization is to constrain the number of factors, or equivalently, the rank of matrix $\mathbf{F}_*$. Such regularized maximum-likelihood estimation can be formulated as an optimization problem

$$\begin{aligned} \max_{\mathbf{F}\in\mathbb{S}_+^M,\sigma^2\geq 0} \quad & \log p(\mathcal{X}|\boldsymbol{\Sigma}) \\ \text{s.t.} \quad & \boldsymbol{\Sigma} = \mathbf{F} + \sigma^2\mathbf{I} \\ & \operatorname{rank}(\mathbf{F}) \leq K \end{aligned} \tag{4.8}$$

where $\mathbb{S}_+^M$ denotes the set of all $M \times M$ positive semidefinite symmetric matrices, and K is the number of factors exogenously specified by the user. Recall that the log likelihood of the observation $\mathcal{X}$ can be written as

$$\log p(\mathcal{X}|\boldsymbol{\Sigma}) = -\frac{N}{2}\left(M\log(2\pi) + \log\det(\boldsymbol{\Sigma}) + \operatorname{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{\text{SAM}})\right), \tag{4.9}$$

where $\boldsymbol{\Sigma}_{\text{SAM}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_{(n)}\mathbf{x}_{(n)}^{\mathrm{T}}$ denotes the sample covariance matrix. Although $\log p(\mathcal{X}|\boldsymbol{\Sigma})$ is not concave in $\boldsymbol{\Sigma}$ and therefore (4.8) is not a convex program, Tipping and Bishop (1999) has shown that its solution can be efficiently computed via PCA. This involves first computing an eigendecomposition of the sample covariance matrix $\boldsymbol{\Sigma}_{\text{SAM}} = \mathbf{B}\mathbf{S}\mathbf{B}^{\mathrm{T}}$, where $\mathbf{B} \in \mathbb{R}^{M\times M}$ is an orthonormal matrix and $\mathbf{S}$ is a diagonal matrix whose diagonal elements are sorted in decreasing order. Our estimate for the residual variance is then given by

$$\hat{\sigma}^2 = \frac{1}{M-K}\sum_{k=K+1}^{M}\mathbf{S}_{k,k}.$$

Furthermore, letting $\mathbf{H}$ be a $K \times K$ diagonal matrix with diagonal entries $\mathbf{H}_{k,k} = \mathbf{S}_{k,k} - \sigma^2$ and $\mathbf{B}_{1:K}$ be the $M \times K$ matrix made up of the first K columns of $\mathbf{B}$,

the estimate for the factor loadings is given by $\hat{\mathbf{F}} = \mathbf{B}_{1:K}\mathbf{H}\mathbf{B}_{1:K}^{\mathrm{T}}$. We will refer to this method as *uniform-residual rank-constrained maximum-likelihood*, and use $\mathbf{\Sigma}_{\mathrm{URM}}^{K} = \hat{\mathbf{F}} + \hat{\sigma}^2\mathbf{I}$ to denote the covariance matrix resulting from this procedure.

In our implementation, we employ a version of cross-validation for the selection of K. Details of the procedure can be found in the appendix. Through selection of K, this procedure arrives at a covariance matrix which we will denote by $\mathbf{\Sigma}_{\mathrm{URM}}$.

Penalizing the Trace

Instead of constraining the rank of factor loadings matrix $\mathbf{F}$, Kao and Van Roy (2011) regularized the maximum-likelihood estimation by introducing a trace penalty term. Specifically, they formulated a convex program by using three facts: first, the log-likelihood function (4.9) is concave in the *inverse* covariance matrix $\mathbf{\Sigma}^{-1}$; second, if $\mathbf{\Sigma} = \mathbf{F} + \sigma^2\mathbf{I}$ with $\mathbf{F} \in \mathbb{S}_+^M$, then the matrix defined by $\mathbf{G} = \sigma^{-2}\mathbf{I} - \mathbf{\Sigma}^{-1}$ is in $\mathbb{S}_+^M$ with $\mathrm{rank}(\mathbf{G}) = \mathrm{rank}(\mathbf{F})$; third, for any $\mathbf{G} \in \mathbb{S}_+^M$, it is a common technique to use $\mathrm{tr}(\mathbf{G})$ as a convex surrogate for $\mathrm{rank}(\mathbf{G})$. These facts together suggest working with inverse covariance matrix $\mathbf{\Sigma}^{-1}$ and penalizing $\mathrm{tr}(\mathbf{G})$ when computing maximum-likelihood estimate, as formerly described by a convex program

$$\begin{aligned} \max_{\mathbf{G}\in\mathbb{S}_+^M, v\geq 0} \quad & \log p(\mathcal{X}|\mathbf{\Sigma}) - \lambda \mathrm{tr}(\mathbf{G}) \\ \text{s.t.} \quad & \mathbf{\Sigma}^{-1} = v\mathbf{I} - \mathbf{G}. \end{aligned} \tag{4.10}$$

Here, the variable v represents the reciprocal of residual variance. We will refer to this method as *uniform-residual trace-penalized maximum-likelihood*, and use $\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda}$ to denote the covariance matrix derived from the optimal solution to this convex program. Similarly to URM, the λ here can be selected by cross-validation.

Kao and Van Roy (2011) also provided an analytical solution to (4.10) using PCA. Their solution is based on soft-thresholding eigenvalues, as defined below.

Definition 4.1. *For all symmetric $M \times M$ matrices, denoted by a set $\mathbb{S}^M$, we define an operator $\mathcal{F}_\lambda : \mathbb{S}^M \to \mathbb{S}^M$ such that $\mathcal{F}_\lambda(\mathbf{A}) = \mathbf{B}$ if $\mathbf{A}$ and $\mathbf{B}$ share the same eigenvectors, and their corresponding eigenvalues, denoted by $a_1, a_2, \ldots, a_M$ and $b_1, b_2, \ldots, b_M$, sum*

to the same trace and satisfy

$$b_i = \max\left\{a_i - \frac{2\lambda}{N}, \frac{1}{v}\right\}, \quad i = 1, \ldots, M,$$

for some scalar v*.*

They have shown that

$$\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}} = \mathcal{F}_{\lambda}(\mathbf{\Sigma}_{\mathrm{SAM}}). \tag{4.11}$$

Therefore, to compute $\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}$, we first compute the eigendecomposition of $\mathbf{\Sigma}_{\mathrm{SAM}}$, and then determine the value of v such that the eigenvalues given by the above formula sum to the desired trace. Note that for reasonably large λ, operator $\mathcal{F}_{\lambda}$ will flatten most eigenvalues of the input matrix and allow only the largest eigenvalues to remain outstanding, which effectively produces an output matrix that is the sum of a low-rank matrix and a multiple of identity matrix, as desired.

One of the main differences between URM and UTM is the way they deal with the large eigenvalues of the sample covariance matrix. While URM preserves those values in its estimate, UTM subtracts a constant $\frac{2\lambda}{N}$ from them. Such subtraction has been shown to correct the bias induced by sample eigenvalues, and generally yields more accurate estimates (Kao and Van Roy, 2011).

4.3.2 Posterior-Constrained Empirical Optimization

Both of the methods discussed above focus on the goodness of fit and ignore the subsequent decision guided by the estimate. We now propose a method that takes into account the decision objective by maximizing the in-sample performance of the resulting decision. Recall from (4.3) that, given an estimate $\mathbf{\Sigma}$, the resulting decision $\mathbf{u}$ can be viewed as a function of it. Let us denote this relation by $\mathbf{u}(\mathbf{\Sigma}) = \frac{1}{2}\mathbf{\Sigma}^{-1}\mathbf{c}$. The *in-sample performance* of $\mathbf{\Sigma}$ is therefore defined as the payoff we receive as if we apply the resulting decision $\mathbf{u}(\mathbf{\Sigma})$ over the observed data, i.e.,

$$\tilde{g}(\mathbf{\Sigma}) = \sum_{n=1}^{N} g\left(\mathbf{u}(\mathbf{\Sigma}), \mathbf{x}_{(n)}\right).$$

A simple *empirical optimization* approach would seek a $\mathbf{\Sigma}$ that maximizes the in-sample performance $\tilde{g}(\mathbf{\Sigma})$. Although this approach explicitly takes into account the decision objective, it generally suffers from over-fitting since the selected estimate is too specialized for in-sample data, and the resulting decision does not generalize well to future samples.

One remedy to this problem is to require $\mathbf{\Sigma}$ to be selected from a set of covariance matrices that are coherent with our model assumption and well explain the historical data. Since posterior probability $p(\mathbf{\Sigma}|\mathcal{X})$ effectively reflects these two criteria, we propose selecting a $\mathbf{\Sigma}$ that maximizes in-sample performance $\tilde{g}(\mathbf{\Sigma})$ subject to a constraint that its posterior probability $p(\mathbf{\Sigma}|\mathcal{X})$ is sufficiently hight, as formerly described by

$$\begin{aligned} \max_{\mathbf{\Sigma}} \quad & \tilde{g}(\mathbf{\Sigma}) \\ \text{s.t.} \quad & p(\mathbf{\Sigma}|\mathcal{X}) \geq p_0. \end{aligned}$$

To select a prior for this purpose, note that the trace penalty introduced in Section 4.3.1 is particularly suitable. Specifically, if we view $\exp(-\lambda \mathrm{tr}(\mathbf{G}))$ as a prior for $\mathbf{\Sigma}$, then $\log p(\mathbf{\Sigma}|\mathcal{X})$ equals to $\log p(\mathcal{X}|\mathbf{\Sigma}) - \lambda \mathrm{tr}(\mathbf{G}) +$ a constant, which is concave in $\mathbf{\Sigma}^{-1}$ and leads to a desirable convex level-set. This observation together with the fact that $\tilde{g}$ is a concave function of $\mathbf{\Sigma}^{-1}$ suggest a convex formulation:

$$\begin{aligned} \max_{\mathbf{G}\in\mathbb{S}_+^M, v\geq 0} \quad & \tilde{g}(\mathbf{\Sigma}) \\ \text{s.t.} \quad & \log p(\mathcal{X}|\mathbf{\Sigma}) - \lambda \mathrm{tr}(\mathbf{G}) \geq \bar{p} - \epsilon \\ & \mathbf{\Sigma}^{-1} = v\mathbf{I} - \mathbf{G} \end{aligned} \tag{4.12}$$

where $\bar{p}$ is the optimum of (4.10), and ϵ effectively specifies the "radius" of the candidate level-set. We will refer to this method as *posterior-constrained empirical optimization*, and denote the resulting estimate by $\mathbf{\Sigma}_{\mathrm{PEO}}^{\lambda,\epsilon}$.

To make this formulation practically useful, we also need an efficient solution. One typical approach for solving (4.12) is to absorb the inequality constraint into the

objective and re-write it as

$$\begin{aligned} \max_{\mathbf{G}\in\mathbb{S}_+^M, v\geq 0} \quad & \gamma\tilde{g}(\mathbf{\Sigma}) + \log p(\mathcal{X}|\mathbf{\Sigma}) - \lambda \mathrm{tr}(\mathbf{G}) \\ \text{s.t.} \quad & \mathbf{\Sigma}^{-1} = v\mathbf{I} - \mathbf{G} \end{aligned} \tag{4.13}$$

where $\gamma \geq 0$ is a scalar that adjusts the weights between the in-sample performance and the posterior probability. It is easy to see that γ increases with ϵ, and $\gamma = 0$ when $\epsilon = 0$. Therefore, to solve (4.12), we can solve (4.13) for a sequence of γ, and pick the solution that corresponds to the largest γ while remaining feasible with respect to (4.12). The problem then boils down to how to solve (4.13) efficiently.

Since (4.13) involves a semidefinite constraint and a trace penalty, one might consider solving the problem through application of an alternating direction method of multipliers (ADMM) (see, e.g., Boyd et al. (2011)). This approach, however, gives rise to onerous computational demands when the data dimension M is large. One contribution of this work is an efficient algorithm that solves (4.13) using a variation of PCA. This method is justified by a theoretical result that relates the decision objective to the principal components of the PEO estimate. Specifically, letting $(\mathbf{G}_\gamma, v_\gamma)$ be the solution to (4.13) and $\mathbf{\Sigma}_\gamma = (v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1}$ be the resulting estimate, we have:

Theorem 4.1. *$\mathbf{\Sigma}_\gamma$ is a fixed point that satisfies*

$$\mathbf{\Sigma}_\gamma = \mathcal{F}_\lambda \left(\mathbf{\Sigma}_{\mathrm{SAM}} + \gamma \mathbf{C} \otimes \mathbf{D} \right), \tag{4.14}$$

where $\mathbf{C} = \mathbf{c}\mathbf{c}^{\mathrm{T}}$, $\mathbf{D} = \mathbf{\Sigma}_\gamma^{-1}\mathbf{\Sigma}_{\mathrm{SAM}} - \mathbf{I}$, and $\mathbf{C} \otimes \mathbf{D} \triangleq \frac{1}{2}(\mathbf{C}\mathbf{D} + \mathbf{D}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}})$.

To interpret this result, note that $\mathbf{C}$ essentially contains the information about the decision objective, $\mathbf{D}$ represents the discrepancy between the estimate and the sample covariance matrix (as $\mathbf{D}$ vanishes when $\mathbf{\Sigma}_\gamma = \mathbf{\Sigma}_{\mathrm{SAM}}$), and the operation $\otimes$ captures the alignment between the objective and the discrepancy. Furthermore, the soft-thresholding operator $\mathcal{F}_\lambda$ produces a covariance matrix that is the sum of a low-rank matrix and a multiple of identity matrix, as desired.

Based on Theorem 4.1, we can design an iterative algorithm for solving (4.13). This algorithm evaluates the right-hand-side of (4.14) at each iteration, and use that

result as the search direction for parameters update. Algorithm 1 describes this procedure in detail. In our experiments, we found this method to typically require two orders of magnitude less compute time than ADMM. For example, it can solve a problem of dimension $M = 500$ within seconds on a personal computer, whereas ADMM requires hours to attain the same level of accuracy. We will refer to this algorithm as *directed PCA*, since the selected components are tailored for the decision objective.

Algorithm 1 Procedure for solving (4.13)
Input: $\mathcal{X}, \mathbf{c}, \lambda, \gamma,$ initial point $\mathbf{\Sigma}_0$
Output: $\mathbf{\Sigma}_\gamma$

$(v\mathbf{I} - \mathbf{G})^{-1} \leftarrow \mathbf{\Sigma}_0$ // initialize $(\mathbf{G}, v)$
repeat
  $\hat{\mathbf{D}} \leftarrow (v\mathbf{I} - \mathbf{G})\mathbf{\Sigma}_{\text{SAM}} - \mathbf{I}$ // evaluate the discrepancy between estimate and sample
  $(\hat{v}\mathbf{I} - \hat{\mathbf{G}})^{-1} \leftarrow \mathcal{F}_\lambda\left(\mathbf{\Sigma}_{\text{SAM}} + \gamma\mathbf{C} \otimes \hat{\mathbf{D}}\right)$ // evaluate the RHS of (4.14) using PCA
  $(\Delta\mathbf{G}, \Delta\mathbf{v}) \leftarrow (\hat{\mathbf{G}}, \hat{\mathbf{v}}) - (\mathbf{G}, \mathbf{v})$ // generate search direction
  Use backtracking line-search to select an appropriate step size $\alpha \in (0, 1]$
  $(\mathbf{G}, \mathbf{v}) \leftarrow (\mathbf{G}, \mathbf{v}) + \alpha(\Delta\mathbf{G}, \Delta\mathbf{v})$
until converge
return $(v\mathbf{I} - \mathbf{G})^{-1}$

4.4 Learning Algorithms: Nonuniform Residual

All of the algorithms discussed in Section 4.3 assume the residual variances are identical for each variable, which is apparently not always the case in practice. We will relax such an assumption in this section and discuss three algorithms that can be viewed as extensions of URM, UTM, and PEO.

4.4.1 Expectation Maximization

Without the assumption of uniform residual variances, the rank-constrained maximum-likelihood method can be written as

$$\begin{aligned} \max_{\mathbf{F}\in\mathbb{S}_+^M,\mathbf{R}\in\mathbb{D}_+^M} \quad & \log p(\mathcal{X}|\boldsymbol{\Sigma}) \\ \text{s.t.} \quad & \boldsymbol{\Sigma} = \mathbf{F} + \mathbf{R} \\ & \operatorname{rank}(\mathbf{F}) \leq K \end{aligned} \tag{4.15}$$

where $\mathbb{D}_+^M$ denotes the set of all $M \times M$ diagonal matrices whose entries are non-negative. Unlike (4.8), this formulation does not have an analytical solution. One common approach to solving it approximately is through the expectation-maximization algorithm (EM). We now give a sketch of this method, while more details can be found in Rubin and Thayer (1982).

The algorithm generates a sequence of iterates $\mathbf{F}^{\frac{1}{2}} \in \mathbb{R}^{M\times K}$ and $\mathbf{R} \in \mathbb{D}_+^M$, such that the covariance matrix $\boldsymbol{\Sigma} = \mathbf{F}^{\frac{1}{2}}\mathbf{F}^{\frac{\mathrm{T}}{2}} + \mathbf{R}$ increases the log-likelihood of $\mathcal{X}$ with each iteration. Each iteration involves an estimation step in which we compute expectations $\mathrm{E}[\mathbf{z}_{(n)}|\mathbf{x}_{(n)}]$ and $\mathrm{E}[\mathbf{z}_{(n)}\mathbf{z}_{(n)}^{\mathrm{T}}|\mathbf{x}_{(n)}]$, for $n = 1, \ldots, N$, assuming the data are generated according to the covariance matrix $\boldsymbol{\Sigma} = \mathbf{F}^{\frac{1}{2}}\mathbf{F}^{\frac{\mathrm{T}}{2}} + \mathbf{R}$. A maximization step then updates $\mathbf{F}$ and $\mathbf{R}$ according to these expectations. This algorithm is guaranteed to converge, though not necessarily to a global optimum.

4.4.2 Rescaled Trace-Penalized Maximum-Likelihood

To relax the uniform residual assumption for trace-penalized maximum-likelihood, Kao and Van Roy (2011) propose a method based on componentwise scaling of the data. Specifically, let $\mathbf{T} \in \mathbb{D}_+^M$ be a rescaling matrix, and let $\mathbf{T}\mathcal{X} = \{\mathbf{T}\mathbf{x}_{(1)}, \ldots, \mathbf{T}\mathbf{x}_{(N)}\}$. It is easy to show that $p(\mathcal{X}|\boldsymbol{\Sigma}) = p(\mathbf{T}\mathcal{X}|\mathbf{T}\boldsymbol{\Sigma}\mathbf{T})$ if $\mathbf{T}$ has unit determinant. This observation motivates an approach that simultaneously seeks an appropriate rescaling

matrix $\mathbf{T}$ and a factor model that best explains the rescaled data, as formerly described by the following optimization problem:

$$\begin{aligned} \max_{\mathbf{G}\in\mathbb{S}_+^M, v\in\mathbb{R}_+, \mathbf{T}\in\mathbb{D}_+^M} \quad & \log p(\mathbf{T}\mathcal{X}|\mathbf{\Sigma}) - \lambda \mathrm{tr}(\mathbf{G}) \\ \text{s.t.} \quad & \mathbf{\Sigma}^{-1} = v\mathbf{I} - \mathbf{G}. \\ & \log\det \mathbf{T} \geq 0. \end{aligned} \tag{4.16}$$

The solution to this problem identifies a linear transformation that allows the data to be best-explained by a factor model with uniform residual variances. Given an optimal solution, $1/\mathbf{T}_{i,i}^2$ should be approximately proportional to the variance of the ith residual, so that normalizing by $\mathbf{T}_{i,i}$ makes residual variances uniform. Note that the optimization problem constrains $\log\det\mathbf{T}$ to be nonnegative rather than zero. This makes the feasible region convex, and this constraint is binding at the optimal solution. Denote the optimal solution to (4.16) by $(\mathbf{G}_*, v_*, \mathbf{T}_*)$. The estimate is thus given by $\mathbf{T}_*^{-1}(v_*\mathbf{I} - \mathbf{G}_*)^{-1}\mathbf{T}_*^{-\mathrm{T}}$.

The objective function of (4.16) is not concave in $(\mathbf{G}, v, \mathbf{T})$, but is biconcave in $(\mathbf{G}, v)$ and $\mathbf{T}$. We solve it by coordinate ascent, alternating between optimizing $(\mathbf{G}, v)$ and $\mathbf{T}$. This procedure is guaranteed convergence. We will refer to this method as *rescaled trace-penalized maximum-likelihood* (RTM).

4.4.3 Rescaled Posterior-Constrained Empirical Optimization

We now extend PEO to deal with nonuniform residuals by a rescaling technique similar to that presented in Section 4.4.2. Note that for any rescaling matrix $\mathbf{T}$, we have

$$\mathbf{c}^{\mathrm{T}}\mathbf{u} - (\mathbf{u}^{\mathrm{T}}\mathbf{x})^2 = (\mathbf{T}\mathbf{c})^{\mathrm{T}}(\mathbf{T}^{-1}\mathbf{u}) - \left((\mathbf{T}^{-1}\mathbf{u})^{\mathrm{T}}\mathbf{T}\mathbf{x}\right)^2.$$

This implies if we rescale the data, objective vector, and decision by the same rescaling matrix, then the resulting performance does not change. Based on this equivalence, we propose *rescaled posterior-constrained empirical optimization* (RPEO), formally described by the following procedure:

1. Use RTM to produce a rescaling matrix $\mathbf{T}$.
2. Rescale the data and objective vector by $\mathbf{T}$.
3. Apply PEO to the rescaled data and objective to produce a rescaled covariance estimate $\tilde{\mathbf{\Sigma}}$.
4. Compute a pre-scaled decision $\tilde{\mathbf{u}} = \frac{1}{2}\tilde{\mathbf{\Sigma}}^{-1}(\mathbf{Tc})$.
5. Output a decision $\hat{\mathbf{u}} = \mathbf{T}\tilde{\mathbf{u}}$.

We will refer to RPEO and PEO collectively as *directed PCA*.

4.5 Computational Experiments

To compare the performance of aforementioned algorithms, we conducted two sets of experiments. The first one uses synthetic data, whereas the second one is based on the real data from S&P 500 stocks returns.

4.5.1 Synthetic Data

For synthetic data experiment, we further divided it into two cases: one with uniform residual variances, and the other with nonuniform residual variances. The former used the following procedure to generate data:

1. Sample M orthonormal vectors $\phi_1, \phi_2, \ldots, \phi_M \in \mathbb{R}^M$ isotropically.
2. Sample $f_1, f_2, \ldots, f_M$ iid from $\mathcal{N}(-1, 2)$.
3. Let $\mathbf{F}_*^{\frac{1}{2}} = [e^{f_1}\phi_1 \quad e^{f_2}\phi_2 \quad \ldots \quad e^{f_M}\phi_M]$.
4. Let $\mathbf{\Sigma}_* = \mathbf{F}_*^{\frac{1}{2}}\mathbf{F}_*^{\frac{\mathrm{T}}{2}} + \mathbf{I}$.
5. Sample $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(N)}$ iid from $\mathcal{N}(0, \mathbf{\Sigma}_*)$.

Note that we sampled the magnitudes of factors from a log-normal distribution, and therefore only a small fraction of them would be significant while the others close to zero. This is intended to simulate the scenario where the covariance matrix can be well approximated by but not exactly equal to the sum of a low-rank matrix and a diagonal one, as in many real-world cases.

We repeated this procedure 100 times for each $N \in \{25, 50, 100, 200\}$ with $M = 100$, and tested URM, UTM, and PEO on these uniform-residual datasets. For URM and UTM, the parameters of the prior distribution K and λ were selected via cross-validation, where about 70% of each dataset was used for training and 30% for validation. For PEO, we chose their λ to be the same value as UTM's, and set their ϵ to be $8, 6, 4, 2$, for $N = 25, 50, 100, 200$, respectively.

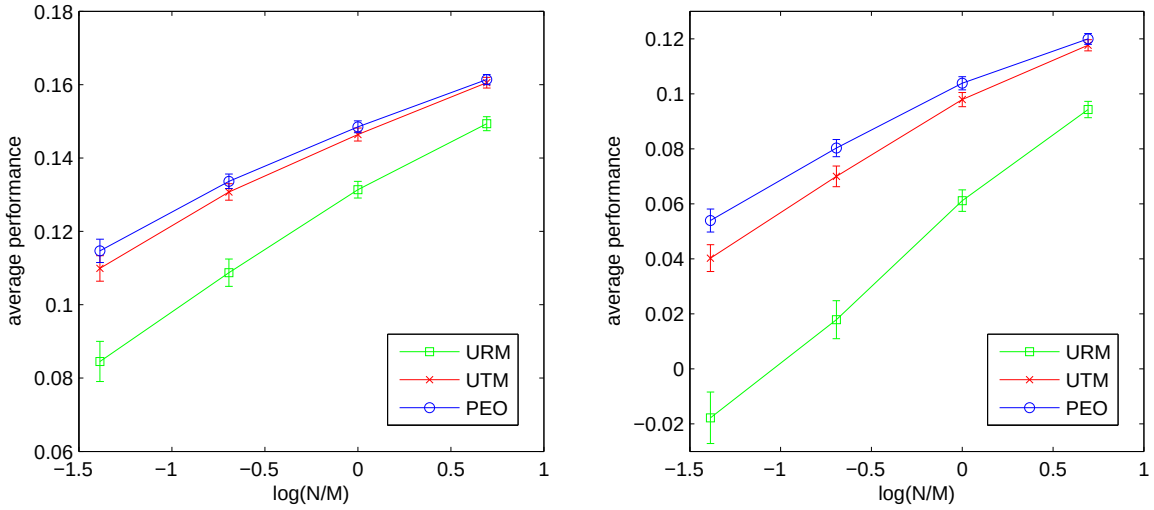
To generate an objective vector $\mathbf{c}$, we considered two cases: independent objective and aligned objective. In the first case, we simply sampled a $\mathbf{c}$ from the unit sphere in $\mathbb{R}^M$ isotropically. In the second case, we generated a $\mathbf{c}$ that was relatively more aligned with the top twenty primary factor loading vectors. Specifically, let us assume without loss of generality that $f_1 > f_2 > \ldots > f_M$. We sampled $p_1, p_2, \ldots, p_{20}$ iid from $\mathcal{N}(0, 4)$, and set $\breve{\mathbf{c}} = \sum_{i=1}^{20} p_i \phi_i + \bar{\mathbf{c}}$, where $\bar{\mathbf{c}}$ was drawn from $\mathcal{N}(0, \mathbf{I}_M)$. This $\breve{\mathbf{c}}$ was then normalized to produce a unit-length $\mathbf{c}$. We can see that such $\mathbf{c}$ vector tends to have larger projections on $\phi_1, \ldots, \phi_{20}$ than other directions. As we shall see in the results, directed PCA has more prominent advantage in such scenario. It is also worth mentioning that such scenario is not unusual in practice. For example, in portfolio management, $\mathbf{c}$ represents the expected return of assets and can often be weakly aligned with some market factors.

To compare the performance, we consider out-of-sample objective value delivered by the decision generated by each algorithm. Specifically, once each algorithm produces a covariance matrix estimate $\hat{\mathbf{\Sigma}}$, we compute a decision $\hat{\mathbf{u}} = \frac{1}{2}\hat{\mathbf{\Sigma}}^{-1}\mathbf{c}$, and then evaluate the out-of-sample objective value by $\mathrm{E}[g(\mathbf{x}, \hat{\mathbf{u}})] = \mathbf{c}^{\mathrm{T}}\hat{\mathbf{u}} - \hat{\mathbf{u}}^{\mathrm{T}}\mathbf{\Sigma}_*\hat{\mathbf{u}}$. Figure 4.1 plots the average out-of-sample objective value delivered by each algorithm for the independent and aligned objective cases. Here the x-axis is the log-ratio of the number of samples to the number of variables. This measure represents the availability of data relative to the number of variables, and is expected to drive performance differences. It is easy to see that PEO has an advantage over UTM, and the advantage is particularly large when the objective vector is weakly aligned with primary factors. Indeed, the gain of PEO over UTM can be as high as 34% when the size of dataset is small.

Our second type of synthetic data was generated using a similar procedure except Step 4 was replaced by

$$\mathbf{\Sigma}_* = \mathbf{F}_*^{\frac{1}{2}} \mathbf{F}_*^{\frac{\mathrm{T}}{2}} + \mathrm{diag}(e^{r_1}, \cdots, e^{r_M}),$$

where $r_1, \ldots, r_M$ were drawn iid from $\mathcal{N}(0, 0.6^2)$. This way we effectively introduced moderate variation into residual variances. EM, RTM, and RPEO were tested on this nonuniform residual dataset, and Figure 4.2 plots the average out-of-sample objective value delivered by these algorithms for both independent and aligned objective vectors. These results has similar trend as in Figure 4.1, except a mild increase in the advantage of RPEO.



(a) (b)

Figure 4.1: The average out-of-sample objective value delivered by URM, UTM, and PEO, for (a) independent objective, and (b) aligned objective.

4.5.2 Real Data

Since portfolio optimization is an important application of directed PCA, in this subsection we present our experiment results for such setting using real stock return

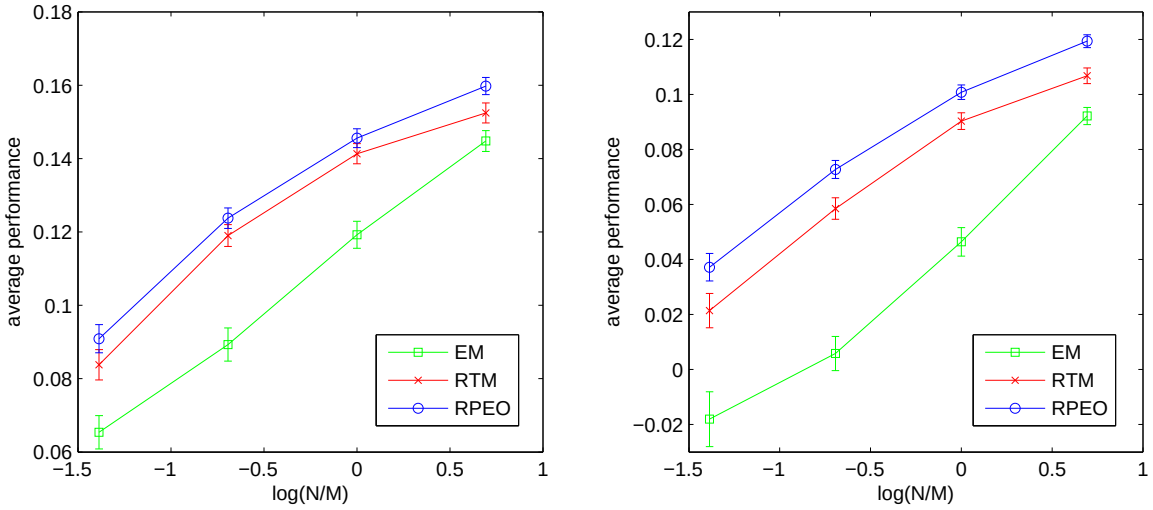


Figure 4.2: The average out-of-sample objective value delivered by EM, RTM, and RPEO, for (a) independent objective, and (b) aligned objective.

data. A typical portfolio optimization problem aims to select a portfolio that maximizes *certain-equivalent payoff*, defined as the expected payoff of the portfolio minus the variance of the payoff weighted by a risk-aversion coefficient. Let us denote the future returns of M assets by a vector $\mathbf{x} \in \mathbb{R}^M$, and denote the dollar amounts we would like to invest on these assets by a vector $\mathbf{u} \in \mathbb{R}^M$. We can write the objective as

$$\max_{\mathbf{u}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{x}] - \xi \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{x}],$$

where the risk-aversion coefficient ξ is usually set to some small value in the order of 10^{-6}. Now suppose we are given a good estimate of the expected future returns $\mathrm{E}[\mathbf{x}]$, denoted by a vector $\hat{\mu} \in \mathbb{R}^M$. In general, the magnitude of $\hat{\mu}$ is much smaller than that of $\mathbf{x}$, and therefore $\mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{x}] \simeq \mathrm{E}[(\mathbf{u}^{\mathrm{T}}\mathbf{x})^2]$. Based on this approximation, we can recast the objective by $\mathrm{E}[g(\mathbf{u}, \mathbf{x})]$, where g is defined in (4.1) with objective vector $\mathbf{c} = \xi^{-1}\hat{\mu}$. We now demonstrate how directed PCA can help generate better portfolio decision $\mathbf{u}$ in this scenario.

Our experiments involve estimation of covariance matrices from historical daily

returns of stocks represented in the S&P 500 index as of March, 2011. We use price data collected from the period starting November 2, 2001, and ending August 9, 2007. This period was chosen to avoid the erratic market behavior observed during the bursting of the dot-com bubble in 2000 and the financial crisis that began in 2008. Daily returns were computed from closing prices while outliers were clipped. Over this duration, there were 1450 trading days, indexed by $1, \ldots, 1450$, and 453 stocks under consideration. Let us denote the daily return of stock i on day t by $y_{i,t}$, and use $\mathbf{y}_{(t)} = [y_{1,t} \quad \ldots \quad y_{M,t}]^{\mathrm{T}}$, $M = 453$, to represent the return vector on day t. More details on this pre-processing procedure can be found in the appendix.

However, these $\mathbf{y}_{(t)}$ vectors can hardly be regarded as stationary, mainly due to the considerable fluctuation of volatility over time. To mitigate such variation, we also computed the 50-day-average volatility at each time point $t > 50$ for every asset i, denoted by $\eta_{i,t}$. We used these quantities to normalize daily returns so that their distribution is closer to being stationary. Because of this normalization, the objective vector $\mathbf{c}$ and decision $\mathbf{u}$ were also rescaled accordingly to compensate this modification, as reasoned in Section 4.4.3.

We generated estimates corresponding to each among a subset of the 1450 days. As would be done in real-time application, for each such day t we used N data points $\{\mathbf{y}_{(t-N+1)}, \ldots, \mathbf{y}_{(t)}\}$ that would have been available on that day to compute the estimate and subsequent data to assess performance. In particular, we generated estimates every 20 days beginning on day 1350 and ending on day 1430. For each of these days, we evaluated the certain-equivalent payoff over the next 20 days. Algorithm 2 formalizes this procedure. For each algorithm $\mathcal{U}$, we took the average of these five 20-day averages to be its out-of-sample performance, defined as

$$\frac{1}{5}\sum_{j=0}^{4} \mathcal{T}(\hat{\mu}, \mathcal{U}, N, 1350 + 20j).$$

In our implementation, we set $\xi = 10^{-6}$, $\epsilon = 40$, and the regularization parameters K and λ were selected by a cross-validation procedure.

To evaluate the efficacy of our method for a wide range of possible scenarios, we repeated the above procedure one hundred times, each with a different expected

Algorithm 2 Testing Procedure $\mathcal{T}$
Input: expected returns $\hat{\mu}$, learning algorithm $\mathcal{U}$, window size N, time point t
Output: average certain-equivalent payoff over test period $t+1, \ldots, t+20$

$\mathcal{X} \leftarrow \left\{ \mathbf{x}_{(\tau)} \middle| \mathbf{x}_{(\tau)} = \begin{bmatrix} \frac{y_{1,\tau}}{\eta_{1,\tau}} & \ldots & \frac{y_{M,\tau}}{\eta_{M,\tau}} \end{bmatrix}^{\mathrm{T}}, \tau = t-N+1, \ldots, t \right\}$ // volatility-normalized returns
$\mathbf{c} \leftarrow \xi^{-1} \begin{bmatrix} \frac{\hat{\mu}_1}{\eta_{1,t}} & \ldots & \frac{\hat{\mu}_M}{\eta_{M,t}} \end{bmatrix}^{\mathrm{T}}$ // rescale objective vector by the current estimate for volatility
$\hat{\Sigma} \leftarrow \mathcal{U}(\mathcal{X}, \mathbf{c})$
$\hat{\mathbf{u}} \leftarrow \frac{1}{2} \hat{\mathbf{\Sigma}}^{-1} \mathbf{c}$
$\mathbf{u} \leftarrow \begin{bmatrix} \eta_{1,t}\hat{\mathbf{u}}_1 & \ldots & \eta_{M,t}\hat{\mathbf{u}}_M \end{bmatrix}^{\mathrm{T}}$ // inversely rescale the decision by the current estimate for volatility
return $\frac{1}{20} \sum_{\tau=t+1}^{t+20} \left(\hat{\mu}^{\mathrm{T}} \mathbf{u} - \xi (\mathbf{u}^{\mathrm{T}} \mathbf{y}_{(\tau)})^2 \right)$

returns vector $\hat{\mu}$ randomly sampled from $\mathcal{N}(0, 10^{-6}\mathbf{I})$. Figure 4.3 plots the average certain-equivalent payoff delivered by EM, RTM, and RPEO for each $N \in \{600, 800, 1000, 1200\}$. RPEO is the dominant solution, generally outperforming the runner-up RTM by 7%. It is worth noting that the performance of each algorithm peaks at $N = 1000$. If the time series were stationary, one would expect performance to monotonically improve with N, as we have observed in the synthetic data experiment. However, this is a real time series and might not be perfectly stationary. We believe that its distribution changes enough over about a thousand trading days so that using historical data collected further back degrades the estimates. This observation indeed suggests that in real applications RPEO is likely to deliver significantly superior performance than RTM or EM even when a large amount of data is provided. This is in contrast with the synthetic data experiment, which may have led to an impression that the performance difference could be made arbitrarily small by using more data.

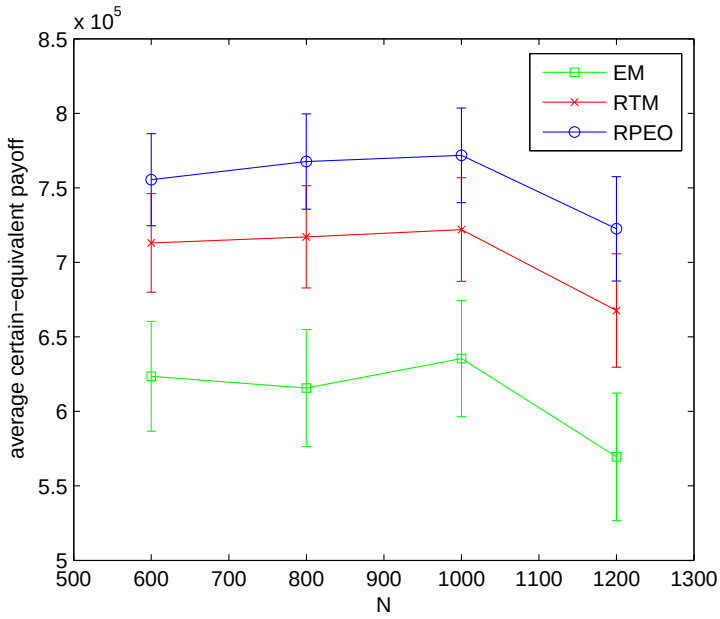


Figure 4.3: The average certain-equivalent payoff delivered by EM, RTM, and RPEO, for 100 randomly generated objectives.

4.6 Analysis

Through computational studies, we have demonstrated that directed PCA leads to better decisions than conventional PCA. In this section we provide an analysis that helps to explain the sources of improvement. We will focus on analysis of the uniform-residual case, though extension to nonuniform-residual case is straightforward. Our analysis will focus on a comparison between PEO and UTM, since PEO can be viewed as a variation of UTM that takes the decision objective into account and both PEO and UTM outperform URM. For further discussion comparing UTM against URM, we refer the reader to Kao and Van Roy (2011).

Let us start by developing some intuition for what directed PCA does. Recall that PEO aims to maximize $\tilde{g}(\mathbf{\Sigma})$ while maintaining high $p(\mathbf{\Sigma}|\mathcal{X})$. Since $\mathbf{\Sigma}_{\text{SAM}}$ maximizes $\tilde{g}(\mathbf{\Sigma})$ and $\mathbf{\Sigma}^{\lambda}_{\text{UTM}}$ maximizes $p(\mathbf{\Sigma}|\mathcal{X})$, we can think of $\mathbf{\Sigma}^{\lambda,\epsilon}_{\text{PEO}}$ as an estimate that deviates from $\mathbf{\Sigma}^{\lambda}_{\text{UTM}}$ towards $\mathbf{\Sigma}_{\text{SAM}}$ in order to increase $\tilde{g}(\mathbf{\Sigma})$. Furthermore, because the value of $\tilde{g}(\mathbf{\Sigma})$ is more sensitive to changes along the direction of $\mathbf{c}\mathbf{c}^{\text{T}}$, such deviations tend

to be larger for components that are aligned with $\mathbf{c}$. The following example illustrates this property.

Suppose

$$\mathbf{\Sigma}_{\text{SAM}} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

and $2\lambda/N = 0.3$. By (4.11), we have

$$\mathbf{\Sigma}_{\text{UTM}} = \begin{bmatrix} 1.7 & 0 \\ 0 & 1.3 \end{bmatrix}.$$

To simplify illustration, let us restrict attention to covariance matrices that take the form $\mathbf{\Sigma}^{-1} = \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix}$. Figure 4.4(a) plots the level sets of $\tilde{g}(\mathbf{\Sigma})$ and $p(\mathbf{\Sigma}|\mathcal{X})$ in the x-y plane, for an objective vector $\mathbf{c} = [0.8 \quad 0.2]^{\text{T}}$ that is closely aligned with $[1 \quad 0]^{\text{T}}$. We label $\mathbf{\Sigma}_{\text{UTM}}$, $\mathbf{\Sigma}_{\text{SAM}}$, and $\mathbf{\Sigma}_{\text{PEO}}$ in the same plot. It is easy to see that the displacement between PEO and SAM is much smaller in the x-direction than in the y-direction. On the other hand, Figure 4.4(b) plots the same contours for an objective vector $\mathbf{c} = [0.2 \quad 0.8]^{\text{T}}$, and in this case PEO deviates from UTM mostly in y-direction.

Let us now move on to establish more formal results. We will consider an idealized, analytically tractable scenario in which the sample covariance matrix $\mathbf{\Sigma}_{\text{SAM}}$ turns out to be identical to $\mathbf{\Sigma}_*$. As we shall see, such simplifying assumption can largely facilitate our analysis and lead to directly interpretable results.

Let an eigen-decomposition of $\mathbf{\Sigma}_*$ be $\mathbf{ALA}^{\text{T}}$, where $\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \ldots \mathbf{a}_M]$ is orthonormal and $\mathbf{L} = \text{diag}(\ell_1, \ell_2, \ldots, \ell_M)$, with $\ell_1 > \ell_2 > \ldots > \ell_K > \ell_{K+1} = \ell_{K+2} = \ldots = \ell_M = \sigma_*^2$. Recall that we evaluate the quality of an estimate $\hat{\mathbf{\Sigma}}$ by the out-of-sample performance of the resulting decision $\frac{1}{2}\hat{\mathbf{\Sigma}}^{-1}\mathbf{c}$. Let us denote such performance measure by a function

$$\mathcal{G}(\hat{\mathbf{\Sigma}}) = \text{E}\left[g\left(\frac{1}{2}\hat{\mathbf{\Sigma}}^{-1}\mathbf{c}, \mathbf{x}\right)\right].$$

Under the simplifying assumption that $\mathbf{\Sigma}_{\text{SAM}} = \mathbf{\Sigma}_*$, we can demonstrate the advantage of PEO over UTM by the following result.

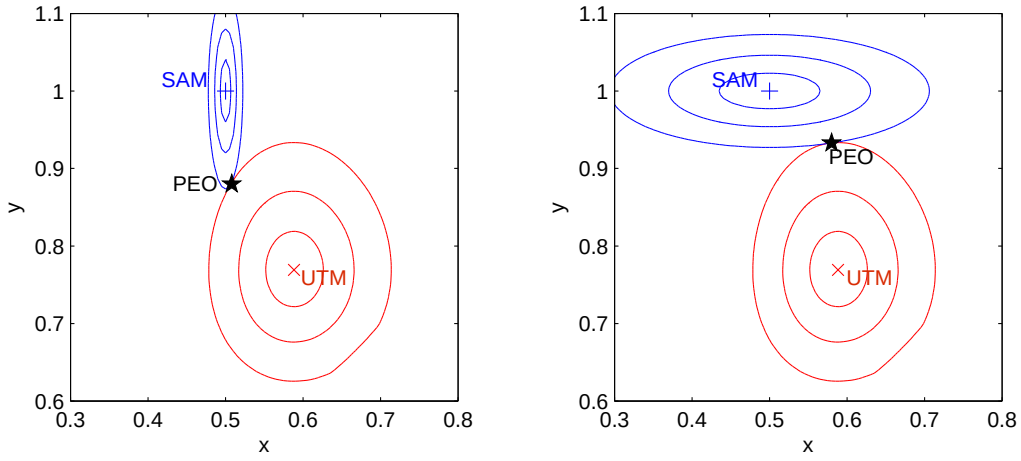


Figure 4.4: The level sets of $p(\boldsymbol{\Sigma}|\mathcal{X})$ is given by the red lines, whereas the blue lines plot the level sets of $\tilde{g}(\boldsymbol{\Sigma})$ for (a) $\mathbf{c} = [0.8 \quad 0.2]^{\mathrm{T}}$ and (b) $\mathbf{c} = [0.2 \quad 0.8]^{\mathrm{T}}$. The red cross, blue plus, and black star represent $\boldsymbol{\Sigma}_{\mathrm{UTM}}$, $\boldsymbol{\Sigma}_{\mathrm{SAM}}$, and $\boldsymbol{\Sigma}_{\mathrm{PEO}}$, respectively.

Proposition 4.1. *If* $\mathbf{c} \in \{\mathbf{a}_1, \ldots, \mathbf{a}_M\}$, *for any* $\lambda > 0$ *such that* $\mathcal{G}\left(\boldsymbol{\Sigma}_{\mathrm{UTM}}^{\lambda}\right) \neq \sup_{\boldsymbol{\Sigma}} \mathcal{G}\left(\boldsymbol{\Sigma}\right)$, *we have*

$$\mathcal{G}\left(\boldsymbol{\Sigma}_{\mathrm{PEO}}^{\lambda,\epsilon}\right) > \mathcal{G}\left(\boldsymbol{\Sigma}_{\mathrm{UTM}}^{\lambda}\right), \quad \forall \epsilon > 0.$$

Furthermore, the gap of this inequality monotonically increases with ϵ.

This proposition indicates that, under the simplifying assumption, if the objective vector is an eigenvector of $\boldsymbol{\Sigma}_*$, and the estimate produced by UTM is not already optimal, then PEO generally outperforms UTM. To further quantify the degree of this improvement, we now give a pair of results that illustrate how the improvement can be arbitrarily large as the data dimension grows. These results will be based on an additional condition on the λ parameter. Specifically, we will assume λ is fixed to $M\sigma_*^2$ from now on. Such setting is recommended by Kao and Van Roy (2011), who has shown through random matrix theory that this parameter choice optimizes the prediction accuracy of UTM estimate in terms of the expected log-likelihood of out-of-sample data.

Using this additional condition, our next result identifies an asymptotic regime

where the improvement of PEO over UTM grows with the data dimension.

Proposition 4.2. *Fixing N, K, and σ_*^2, consider a sequence of covariance matrices $\mathbf{\Sigma}_*^{(M)}$ and objective vectors $\mathbf{c}^{(M)}$, indexed by the dimension M, that satisfy $\ell_i^{(M)} \in [\frac{2\lambda}{N} - \delta_i, \frac{2\lambda}{N} + \delta_i]$ for $i = 1, 2, \ldots, K$ and constants $\delta_1, \ldots, \delta_K$. If $\mathbf{c}^{(M)} \in \left\{\mathbf{a}_1^{(M)}, \mathbf{a}_2^{(M)}, \ldots, \mathbf{a}_K^{(M)}\right\}$, we have*

$$\lim_{\epsilon \to \infty} \mathcal{G}(\mathbf{\Sigma}_{\text{PEO}}^{\lambda,\epsilon}) - \mathcal{G}(\mathbf{\Sigma}_{\text{UTM}}^{\lambda}) = \Omega(M).$$

The above result implies the performance gap between PEO and UTM is particularly large when the objective vector is aligned with the factor loading vectors, as we have observed in our experiment results. For the opposite case where the objective vector is orthogonal to the factor loading vectors, we can still illustrate the advantage of PEO over UTM in terms of performance ratio, as formerly described below.

Proposition 4.3. *Fixing N and σ_*^2, consider a sequence of covariance matrices $\mathbf{\Sigma}_*^{(M)}$ and objective vectors $\mathbf{c}^{(M)}$, indexed by the dimension M, that satisfy $K^{(M)} = \lceil \alpha M \rceil$, where $\alpha \in (0, 1)$ is a constant, and $\ell_i^{(M)} > \frac{2\lambda}{N} + \sigma_*^2$ for $i = 1, 2, \ldots, K^{(M)}$. If $\mathbf{c}^{(M)} \perp \text{span}\left\{\mathbf{a}_1^{(M)}, \mathbf{a}_2^{(M)}, \ldots, \mathbf{a}_{K^{(M)}}^{(M)}\right\}$, we have*

$$\lim_{\epsilon \to \infty} \frac{\mathcal{G}(\mathbf{\Sigma}_{\text{PEO}}^{\lambda,\epsilon})}{\mathcal{G}(\mathbf{\Sigma}_{\text{UTM}}^{\lambda})} = \Omega(M).$$

This pair of results together suggest that directed PCA generally offers substantial improvement over conventional PCA methods for high-dimensional problems, and indeed manifest the importance of accounting for the decision objective in the estimation process.

4.7 Summary

We have proposed a new approach to covariance matrix estimation, which we refer to as *directed PCA*. The idea is to produce a covariance matrix estimate that corresponds to a factor model with factor loadings and residual variances estimated in a

way that optimizes in-sample performance of the resulting decision strategy subject to a constraint that the model explains the data well. Such a method effectively incorporates the decision objective into the model fitting procedure. Computational and theoretical analyses demonstrate that our approach indeed outperforms conventional methods.

There is a growing body of research on variations of PCA, including methods that generate sparse factor loadings (Jolliffe et al., 2003; Zou et al., 2004; D'Aspremont et al., 2004; Johnstone and Lu, 2007; Amini and Wainwright, 2009) and methods that are resistant to corrupted data (Pison et al., 2003; Candès et al., 2009; Xu et al., 2010). It would be interesting to explore how to incorporate decision objectives into those settings.

Chapter 5

Conclusion

When applying machine learning for decision making, practitioners often treat model estimation and decision optimization independently, resulting in suboptimal solutions. In this research we have proposed the concept of directed learning, and developed three algorithms, directed regression, directed time-series regression, and directed PCA, all of which use the decision objective to guide model fitting. Through computational studies and theoretical analysis, we have demonstrated that these algorithms greatly improve decision performance over conventional methods.

Although we only cover three problem settings in this thesis, our work suggests that there can be significant gains in broader problem contexts from a tighter coupling between machine learning and decision making. In particular, machine learning algorithms should factor decision objectives into the learning process. It will be interesting to explore how to do this with other classes of models and objectives.

Let us close by mentioning that the ideas behind directed learning ought to play a role in reinforcement learning as presented in Sutton and Barto (1998). Reinforcement learning algorithms learn from experience to predict a sum of future rewards as a function of a state, typically by fitting a linear combination of features of the state. This so-called approximate value function is then used to guide sequential decision making. Understanding how to synthesize directed learning with reinforcement learning is an interesting direction for future research.

Appendix A

Proofs

Theorem 2.1. *For all N, M, K, μ_Q, σ_w, and $\overline{\mathcal{O}}$,*

$$\hat{\mathbf{r}}_0 = \underset{\mathbf{r}\in\mathbb{R}^K}{\operatorname{argmin}} \sum_{n=1}^{N} \left\|\mathbf{y}^{(n)} - \mathbf{X}^{(n)}\mathbf{r}\right\|^2$$

and

$$\hat{\mathbf{r}}_\infty = \underset{\mathbf{r}\in\mathbb{R}^K}{\operatorname{argmin}} \sum_{n=1}^{N} \ell(\mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)}), \mathbf{y}^{(n)}).$$

Proof. Let

$$\breve{\mathbf{X}} = \begin{bmatrix} \mathbf{X}^{(1)} \\ \vdots \\ \mathbf{X}^{(N)} \end{bmatrix}, \breve{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z}^{(1)} \\ \vdots \\ \mathbf{Z}^{(N)} \end{bmatrix}, \breve{\mathbf{y}} = \begin{bmatrix} \mathbf{y}^{(1)} \\ \vdots \\ \mathbf{y}^{(N)} \end{bmatrix}.$$

Let $\bar{\mathbf{r}} = \mathrm{E}[\mathbf{r}^*|\overline{\mathcal{O}}]$, $\bar{\mathbf{r}}^\perp = \mathrm{E}[\mathbf{r}^{\perp *}|\overline{\mathcal{O}}]$. For any matrix $\mathbf{V}$, let $\mathbf{V}^\dagger$ denote the Moore-Penrose pseudoinverse $(\mathbf{V}^{\mathrm{T}}\mathbf{V})^{-1}\mathbf{V}^{\mathrm{T}}$. Recall that $\langle \mathbf{C}_k, \tilde{\mathbf{D}}_j \rangle = 0, \forall\, k, j$ implies that each column of $\breve{\mathbf{X}}$ is orthogonal to each column of $\breve{\mathbf{Z}}$. Because $\mathbf{r}^*, \mathbf{r}^{\perp *}, \overline{\mathcal{O}}$ are jointly Gaussian, as $\sigma_r \to \infty$, we have

$$\begin{bmatrix} \bar{\mathbf{r}} \\ \bar{\mathbf{r}}^\perp \end{bmatrix} = \underset{\mathbf{r},\mathbf{r}^\perp}{\operatorname{argmin}} \frac{1}{2\sigma_w^2} \sum_{n=1}^{N} \left\|\mathbf{y}^{(n)} - \mathbf{X}^{(n)}\mathbf{r} - \mathbf{Z}^{(n)}\mathbf{r}^\perp\right\|^2 + \frac{1}{2\sigma_\epsilon^2}\|\mathbf{r}^\perp\|_2^2$$

$$= \underset{\mathbf{r},\mathbf{r}^\perp}{\operatorname{argmin}} \left\| \begin{bmatrix} \frac{1}{\sigma_w}\breve{\mathbf{y}} \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{\sigma_w}\breve{\mathbf{X}} & \frac{1}{\sigma_w}\breve{\mathbf{Z}} \\ 0 & \frac{1}{\sigma_\epsilon}\mathbf{I}_J \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{r}^\perp \end{bmatrix} \right\|^2 = \begin{bmatrix} (\breve{\mathbf{X}}^{\mathrm{T}}\breve{\mathbf{X}})^{-1}\breve{\mathbf{X}}^{\mathrm{T}}\breve{\mathbf{y}} \\ (\breve{\mathbf{Z}}^{\mathrm{T}}\breve{\mathbf{Z}} + \frac{\sigma_w^2}{\sigma_\epsilon^2}\mathbf{I})^{-1}\breve{\mathbf{Z}}^{\mathrm{T}}\breve{\mathbf{y}} \end{bmatrix}.$$

Let $\mathbf{A}^{(n)} = \mathbf{G}_1^{-\frac{1}{2}}\mathbf{G}_2\mathbf{X}^{(n)}$, $\mathbf{B}^{(n)} = \mathbf{G}_1^{-\frac{1}{2}}\mathbf{G}_2\mathbf{Z}^{(n)}$, $\breve{\mathbf{A}} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \vdots \\ \mathbf{A}^{(N)} \end{bmatrix}$, $\breve{\mathbf{B}} = \begin{bmatrix} \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{B}^{(N)} \end{bmatrix}$. We have

$$\begin{aligned}
\hat{\mathbf{r}} &= \underset{\mathbf{r}}{\operatorname{argmin}}\, \mathrm{E}[\ell(\mathbf{u}_{\mathbf{r}}(\tilde{\mathbf{X}}), \tilde{\mathbf{y}})|\overline{\mathcal{O}}] = \underset{\mathbf{r}}{\operatorname{argmin}} \frac{1}{N}\sum_{n=1}^{N} \mathrm{E}_{\tilde{\mathbf{y}}}[\ell(\mathbf{u}_{\mathbf{r}}(\tilde{\mathbf{X}}), \tilde{\mathbf{y}})|\tilde{\mathbf{X}} = \mathbf{X}^{(n)}, \overline{\mathcal{O}}] \\
&= \underset{\mathbf{r}}{\operatorname{argmin}} \sum_{n=1}^{N} \mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)})^{\mathrm{T}}\mathbf{G}_1\mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)}) + \mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)})^{\mathrm{T}}\mathbf{G}_2\, \mathrm{E}[\tilde{\mathbf{y}}|\tilde{\mathbf{X}} = \mathbf{X}^{(n)}, \overline{\mathcal{O}}] \\
&= \underset{\mathbf{r}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{4}\mathbf{r}^{\mathrm{T}}\mathbf{A}^{(n)\mathrm{T}}\mathbf{A}^{(n)}\mathbf{r} - \frac{1}{2}\mathbf{r}^{\mathrm{T}}\mathbf{A}^{(n)\mathrm{T}}(\mathbf{A}^{(n)}\bar{\mathbf{r}} + \mathbf{B}^{(n)}\bar{\mathbf{r}}^{\perp}) \\
&= \bar{\mathbf{r}} + \breve{\mathbf{A}}^{\dagger}\breve{\mathbf{B}}\bar{\mathbf{r}}^{\perp} = \breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} + \breve{\mathbf{A}}^{\dagger}\breve{\mathbf{B}}\left(\breve{\mathbf{Z}}^{\mathrm{T}}\breve{\mathbf{Z}} + \frac{\sigma_w^2}{\sigma_\epsilon^2}\mathbf{I}\right)^{-1}\breve{\mathbf{Z}}^{\mathrm{T}}\breve{\mathbf{y}}. \qquad \text{(A.1)}
\end{aligned}$$

Taking $\sigma_\epsilon \to 0$ and $\sigma_\epsilon \to \infty$ yields

$$\hat{\mathbf{r}}_0 = \breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}}, \qquad \text{(A.2)}$$

$$\hat{\mathbf{r}}_\infty = \breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} + \breve{\mathbf{A}}^{\dagger}\breve{\mathbf{B}}\breve{\mathbf{Z}}^{\dagger}\breve{\mathbf{Y}}. \qquad \text{(A.3)}$$

The first part of the theorem then follows because

$$\hat{\mathbf{r}}_0 = \breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} = \underset{\mathbf{r}}{\operatorname{argmin}} \left\|\breve{\mathbf{y}} - \breve{\mathbf{X}}\mathbf{r}\right\|^2 = \underset{\mathbf{r}}{\operatorname{argmin}} \sum_{n=1}^{N} \left\|\mathbf{y}^{(n)} - \mathbf{X}^{(n)}\mathbf{r}\right\|^2.$$

We now prove the second part. Note that

$$\begin{aligned}
&\underset{\mathbf{r}}{\operatorname{argmin}} \sum_{n=1}^{N} \ell(\mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)}), \mathbf{y}^{(n)}) \\
= &\underset{\mathbf{r}}{\operatorname{argmin}} \sum_{n=1}^{N} \mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)})^{\mathrm{T}}\mathbf{G}_1\mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)}) + \mathbf{u}_{\mathbf{r}}(\mathbf{X}^{(n)})^{\mathrm{T}}\mathbf{G}_2\mathbf{y}^{(n)} \\
= &\underset{\mathbf{r}}{\operatorname{argmin}}\, \mathbf{r}^{\mathrm{T}}\breve{\mathbf{A}}^{\mathrm{T}}\breve{\mathbf{A}}\mathbf{r} - 2\mathbf{r}^{\mathrm{T}}\sum_{n=1}^{N}\mathbf{H}^{(n)\mathrm{T}}\mathbf{y}^{(n)} = (\breve{\mathbf{A}}^{\mathrm{T}}\breve{\mathbf{A}})^{-1}\breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{y}},
\end{aligned}$$

where $\mathbf{H}^{(n)} = \mathbf{G}_2^{\mathrm{T}}\mathbf{G}_1^{-1}\mathbf{G}_2\mathbf{X}^{(n)}$ and $\breve{\mathbf{H}} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(N)} \end{bmatrix}$. Each kth column of $\breve{\mathbf{H}}$, denoted by

$$\breve{\mathbf{h}}_k = \begin{bmatrix} \mathbf{G}_2^{\mathrm{T}}\mathbf{G}_1^{-1}\mathbf{G}_2\mathbf{C}_k\phi^{(1)} \\ \vdots \\ \mathbf{G}_2^{\mathrm{T}}\mathbf{G}_1^{-1}\mathbf{G}_2\mathbf{C}_k\phi^{(N)} \end{bmatrix},$$

is in span$\{\mathrm{col}\ \breve{\mathbf{X}}, \mathrm{col}\ \breve{\mathbf{Z}}\}$ because

$$\mathbf{G}_2^{\mathrm{T}}\mathbf{G}_1^{-1}\mathbf{G}_2\mathbf{C}_k \in \mathbb{R}^{M\times Q} = \mathrm{span}\{\mathbf{C}_1, \cdots, \mathbf{C}_K\tilde{\mathbf{D}}_1, \cdots, \tilde{\mathbf{D}}_J\}.$$

Since the residual $\breve{\mathbf{y}}' = \breve{\mathbf{y}} - \breve{\mathbf{X}}\breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} - \breve{\mathbf{Z}}\breve{\mathbf{Z}}^{\dagger}\breve{\mathbf{y}}$ upon projecting $\breve{\mathbf{y}}$ onto span $\{\mathrm{col}\ \breve{\mathbf{X}}, \mathrm{col}\ \breve{\mathbf{Z}}\}$ is orthogonal to the subspace, we have $\breve{\mathbf{h}}_k^{\mathrm{T}}\breve{\mathbf{y}}' = 0, \forall\, k$ and hence $\breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{y}}' = 0$. This implies $\breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{y}} = \breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{X}}\breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} + \breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{Z}}\breve{\mathbf{Z}}^{\dagger}\breve{\mathbf{y}}$. Further, since $\mathbf{A}^{(n)\mathrm{T}}\mathbf{A}^{(n)} = \mathbf{H}^{(n)\mathrm{T}}\mathbf{X}^{(n)}$ and $\mathbf{A}^{(n)\mathrm{T}}\mathbf{B}^{(n)} = \mathbf{H}^{(n)\mathrm{T}}\mathbf{Z}^{(n)}, \forall\, n$, we have

$$\begin{aligned} \hat{\mathbf{r}}_{\infty} &= \breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} + \breve{\mathbf{A}}^{\dagger}\breve{\mathbf{B}}\breve{\mathbf{Z}}^{\dagger}\breve{\mathbf{y}} = (\breve{\mathbf{A}}^{\mathrm{T}}\breve{\mathbf{A}})^{-1}\left(\breve{\mathbf{A}}^{\mathrm{T}}\breve{\mathbf{A}}\breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} + \breve{\mathbf{A}}^{\mathrm{T}}\breve{\mathbf{B}}\breve{\mathbf{Z}}^{\dagger}\breve{\mathbf{y}}\right) \\ &= (\breve{\mathbf{A}}^{\mathrm{T}}\breve{\mathbf{A}})^{-1}\left(\breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{X}}\breve{\mathbf{X}}^{\dagger}\breve{\mathbf{y}} + \breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{Z}}\breve{\mathbf{Z}}^{\dagger}\breve{\mathbf{y}}\right) = (\breve{\mathbf{A}}^{\mathrm{T}}\breve{\mathbf{A}})^{-1}\breve{\mathbf{H}}^{\mathrm{T}}\breve{\mathbf{y}} \end{aligned}$$

as desired.

□

Theorem 2.2. *For all N, M, K, μ_Q, σ_w, σ_ϵ, and $\overline{\mathcal{O}}$,*

$$\hat{\mathbf{r}} = (1-\lambda)\hat{\mathbf{r}}_0 + \lambda\hat{\mathbf{r}}_{\infty},$$

where $\lambda = \frac{N\sigma_\epsilon^2}{N\sigma_\epsilon^2+\sigma_w^2}$.

Proof. Because $\langle\tilde{\mathbf{D}}_i, \tilde{\mathbf{D}}_j\rangle = \mathbf{1}\{i=j\}$, we have $\breve{\mathbf{Z}}^{\mathrm{T}}\breve{\mathbf{Z}} = N\mathbf{I}$. Plugging this into $(A.1)$ and comparing the resultant expression with $(A.2)$ and $(A.3)$ yield the desired result.

□

Theorem 4.1. *$\boldsymbol{\Sigma}_\gamma$ is a fixed point that satisfies*

$$\boldsymbol{\Sigma}_\gamma = \mathcal{F}_\lambda\left(\boldsymbol{\Sigma}_{\text{SAM}} + \gamma \mathbf{C} \otimes \mathbf{D}\right),$$

where $\mathbf{C} = \mathbf{c}\mathbf{c}^{\text{T}}$, $\mathbf{D} = \boldsymbol{\Sigma}_\gamma^{-1}\boldsymbol{\Sigma}_{\text{SAM}} - \mathbf{I}$, *and* $\mathbf{C} \otimes \mathbf{D} \triangleq \frac{1}{2}(\mathbf{C}\mathbf{D} + \mathbf{D}^{\text{T}}\mathbf{C}^{\text{T}})$.

Proof. Expanding $\tilde{g}(\boldsymbol{\Sigma})$ and $\log p(\mathcal{X}|\boldsymbol{\Sigma})$, we can re-write (4.13) as

$$\begin{aligned}
\min_{\mathbf{G},v} \quad & \gamma\left(\frac{1}{2}\mathbf{c}^{\text{T}}(v\mathbf{I}-\mathbf{G})\boldsymbol{\Sigma}_{\text{SAM}}(v\mathbf{I}-\mathbf{G})\mathbf{c} - \mathbf{c}^{\text{T}}(v\mathbf{I}-\mathbf{G})\mathbf{c}\right) - \log\det\left(v\mathbf{I}-\mathbf{G}\right) \\
& + \text{tr}\left(\left(v\mathbf{I}-\mathbf{G}\right)\boldsymbol{\Sigma}_{\text{SAM}}\right) + \lambda' \text{tr}(\mathbf{G}) \\
\text{s.t.} \quad & \mathbf{G} \in \mathbb{S}_+^M
\end{aligned}$$

where $\lambda' = \frac{2\lambda}{N}$. Note that the constraint $v \geq 0$ is implied in the domain of the objective function. We associate a Lagrange multiplier $\boldsymbol{\Omega} \in \mathbb{S}_+^M$ with the $\mathbf{G} \in \mathbb{S}_+^M$ constraint and write down the Lagrangian as

$$\begin{aligned}
\mathcal{L}(\mathbf{G}, v, \boldsymbol{\Omega}) \;=\; & \gamma\left(\frac{1}{2}\mathbf{c}^{\text{T}}(v\mathbf{I}-\mathbf{G})\boldsymbol{\Sigma}_{\text{SAM}}(v\mathbf{I}-\mathbf{G})\mathbf{c} - \mathbf{c}^{\text{T}}(v\mathbf{I}-\mathbf{G})\mathbf{c}\right) - \log\det\left(v\mathbf{I}-\mathbf{G}\right) \\
& + \text{tr}\left(\left(v\mathbf{I}-\mathbf{G}\right)\boldsymbol{\Sigma}_{\text{SAM}}\right) + \lambda' \text{tr}(\mathbf{G}) - \text{tr}(\boldsymbol{\Omega}\mathbf{G}).
\end{aligned}$$

Let $(\mathbf{G}_\gamma, v_\gamma)$ be the optimal solution, and let $\boldsymbol{\Omega}_\gamma$ be the corresponding Lagrangian multiplier. By KKT conditions we have:

$$\begin{aligned}
\nabla_{\mathbf{G}}\mathcal{L}\Big|_{\mathbf{G}_\gamma, v_\gamma, \boldsymbol{\Omega}_\gamma} \;=\; & \gamma\left(\mathbf{c}\mathbf{c}^{\text{T}} - \frac{1}{2}\mathbf{c}\mathbf{c}^{\text{T}}(v_\gamma\mathbf{I}-\mathbf{G}_\gamma)\boldsymbol{\Sigma}_{\text{SAM}} - \frac{1}{2}\boldsymbol{\Sigma}_{\text{SAM}}(v_\gamma\mathbf{I}-\mathbf{G}_\gamma)\mathbf{c}\mathbf{c}^{\text{T}}\right) \\
& + (v_\gamma\mathbf{I}-\mathbf{G}_\gamma)^{-1} - \boldsymbol{\Sigma}_{\text{SAM}} + \lambda'\mathbf{I} - \boldsymbol{\Omega}_\gamma = 0 \qquad \text{(A.4)} \\
\frac{\partial\mathcal{L}}{\partial v}\Big|_{\mathbf{G}_\gamma, v_\gamma, \boldsymbol{\Omega}_\gamma} \;=\; & -\gamma\text{tr}\left(\mathbf{c}\mathbf{c}^{\text{T}} - \frac{1}{2}\mathbf{c}\mathbf{c}^{\text{T}}(v_\gamma\mathbf{I}-\mathbf{G}_\gamma)\boldsymbol{\Sigma}_{\text{SAM}} - \frac{1}{2}\boldsymbol{\Sigma}_{\text{SAM}}(v_\gamma\mathbf{I}-\mathbf{G}_\gamma)\mathbf{c}\mathbf{c}^{\text{T}}\right) \\
& + \text{tr}(\boldsymbol{\Sigma}_{\text{SAM}}) - \text{tr}\left((v_\gamma\mathbf{I}-\mathbf{G}_\gamma)^{-1}\right) = 0 \qquad \text{(A.5)} \\
\boldsymbol{\Omega}_\gamma, \mathbf{G}_\gamma \;\in\; & \mathbb{S}_+^M \qquad \text{(A.6)} \\
\text{tr}(\boldsymbol{\Omega}_\gamma\mathbf{G}_\gamma) \;=\; & 0. \qquad \text{(A.7)}
\end{aligned}$$

Using the fact that $\boldsymbol{\Sigma}_\gamma = (v_\gamma\mathbf{I}-\mathbf{G}_\gamma)^{-1}$ and $\mathbf{D} = \boldsymbol{\Sigma}_\gamma^{-1}\boldsymbol{\Sigma}_{\text{SAM}} - \mathbf{I}$, we can rewrite (A.4)

as

$$-\gamma \mathbf{C} \otimes \mathbf{D} + (v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1} - \mathbf{\Sigma}_{\text{SAM}} + \lambda' \mathbf{I} - \mathbf{\Omega}_\gamma = 0$$

and rewrite (A.5) as

$$\text{tr}\left(\gamma \mathbf{C} \otimes \mathbf{D} + \mathbf{\Sigma}_{\text{SAM}} - (v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1}\right) = 0.$$

To simplify notation, let us define $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma}_{\text{SAM}} + \gamma \mathbf{C} \otimes \mathbf{D}$, and further rewrite (A.4) and (A.5) as

$$\hat{\mathbf{\Sigma}} = (v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1} + \lambda' \mathbf{I} - \mathbf{\Omega}_\gamma \tag{A.8}$$

$$\text{tr}(\hat{\mathbf{\Sigma}}) = \text{tr}\left((v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1}\right) \tag{A.9}$$

Let an eigendecomposition of $\mathbf{G}_\gamma$ be $\mathbf{AQA}^{\text{T}}$ for which $\mathbf{A} = [\mathbf{a}_1 \quad \dots \quad \mathbf{a}_M]$ is orthonormal. Plugging this into (A.7) we get

$$0 = \text{tr}(\mathbf{\Omega}_\gamma \mathbf{G}_\gamma) = \text{tr}(\mathbf{\Omega}_\gamma \mathbf{AQA}^{\text{T}}) = \text{tr}(\mathbf{A}^{\text{T}} \mathbf{\Omega}_\gamma \mathbf{AQ}) = \sum_{i=1}^{M} \mathbf{Q}_{i,i} \mathbf{a}_i^{\text{T}} \mathbf{\Omega}_\gamma \mathbf{a}_i.$$

By (A.6), $\mathbf{Q}_{i,i} \geq 0$ and $\mathbf{a}_i^{\text{T}} \mathbf{\Omega}_\gamma \mathbf{a}_i \geq 0$, $\forall i = 1, \dots, M$, which implies

$$\mathbf{a}_i^{\text{T}} \mathbf{\Omega}_\gamma \mathbf{a}_i = 0 \text{ if } \mathbf{Q}_{i,i} > 0, \quad \forall i = 1, \dots, M.$$

Let $\mathcal{I}_+ = \{i : \mathbf{Q}_{i,i} > 0\}$. Since $\mathbf{\Omega}_\gamma$ is positive semidefinite, for all $i_0 \in \mathcal{I}_+$ we also have $\mathbf{\Omega}_\gamma \mathbf{a}_{i_0} = 0$. Furthermore, since

$$(v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1} = \mathbf{A} \text{diag}\left(\frac{1}{v_\gamma - \mathbf{Q}_{1,1}}, \dots, \frac{1}{v_\gamma - \mathbf{Q}_{M,M}}\right) \mathbf{A}^{\text{T}},$$

multiplying (A.8) by $\mathbf{a}_{i_0}$ leads to

$$\hat{\mathbf{\Sigma}} \mathbf{a}_{i_0} = \frac{\mathbf{a}_{i_0}}{v_\gamma - \mathbf{Q}_{i_0,i_0}} + \lambda' \mathbf{a}_{i_0} = \left(\frac{1}{v_\gamma - \mathbf{Q}_{i_0,i_0}} + \lambda'\right) \mathbf{a}_{i_0} \tag{A.10}$$

which shows $\mathbf{a}_{i_0}$ is an eigenvector of $\hat{\mathbf{\Sigma}}$. Therefore, without loss of generality we

can choose $\mathbf{A}$ to be the eigenvectors of $\hat{\mathbf{\Sigma}}$, and write an eigendecomposition of $\hat{\mathbf{\Sigma}}$ as $\mathbf{ASA}^{\mathrm{T}}$. Comparing this with (A.10), we have

$$\mathbf{S}_{i_0,i_0} = \frac{1}{v_\gamma - \mathbf{Q}_{i_0,i_0}} + \lambda', \quad \forall i_0 \in \mathcal{I}_+. \tag{A.11}$$

Recall that $\mathbf{\Sigma}_\gamma = (v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1} = (v_\gamma \mathbf{I} - \mathbf{AQA}^{\mathrm{T}})^{-1} = \mathbf{AHA}^{\mathrm{T}}$, where $\mathbf{H}_{i,i} = \frac{1}{v_\gamma - \mathbf{Q}_{i,i}}$, for $i = 1, \ldots, M$. Comparing this expression with (A.11) we arrive at

$$\mathbf{H}_{i_0,i_0} = \mathbf{S}_{i_0,i_0} - \lambda', \quad \forall i_0 \in \mathcal{I}_+,$$

or more generally

$$\mathbf{H}_{i,i} = \begin{cases} \mathbf{S}_{i,i} - \lambda' & \text{if } \mathbf{Q}_{i,i} > 0 \\ \frac{1}{v_\gamma} & \text{otherwise} \end{cases}, \quad i = 1, \ldots, M.$$

Since $\mathbf{H}_{i,i} \geq \frac{1}{v_\gamma}$, to see $\mathbf{H}_{i,i} = \max\left\{\mathbf{S}_{i,i} - \lambda', \frac{1}{v_\gamma}\right\}$, it remains to show $\mathbf{H}_{i,i} \geq \mathbf{S}_{i,i} - \lambda'$ for all i. This follows by rearranging (A.8)

$$\begin{aligned} & (v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1} - \hat{\mathbf{\Sigma}} + \lambda' \mathbf{I} = \mathbf{\Omega}_\gamma \succeq 0 \\ \Rightarrow \quad & \mathbf{AHA}^{\mathrm{T}} - \mathbf{ASA}^{\mathrm{T}} + \mathbf{A}\lambda' \mathbf{I} \mathbf{A}^{\mathrm{T}} \succeq 0 \\ \Rightarrow \quad & \mathbf{H} - \mathbf{S} + \lambda' \mathbf{I} \succeq 0 \Rightarrow \mathbf{H}_{i,i} \geq \mathbf{S}_{i,i} - \lambda', \quad \forall i = 1, \ldots, M. \end{aligned}$$

Finally, by (A.9) we know $\hat{\mathbf{\Sigma}}$ and $\mathbf{\Sigma}_\gamma$ share the same trace, and thus $\mathcal{F}_\lambda(\hat{\mathbf{\Sigma}}) = \mathbf{\Sigma}_\gamma$, as desired.

□

To prove Proposition 4.1, we first prove the following lemma.

Lemma A.1. *If* $\mathbf{c}$ *is an eigenvector of* $\mathbf{\Sigma}_{\mathrm{SAM}}$*, then* $\mathbf{\Sigma}^{\lambda,\epsilon}_{\mathrm{PEO}}$ *and* $\mathbf{\Sigma}_{\mathrm{SAM}}$ *share the same eigenvectors.*

Proof. Recall that with appropriate choice of $\gamma \geq 0$, we have $\mathbf{\Sigma}^{\lambda,\epsilon}_{\mathrm{PEO}} = (v_\gamma \mathbf{I} - \mathbf{G}_\gamma)^{-1}$, where $(\mathbf{G}_\gamma, v_\gamma)$ is the solution to (4.13). One standard approach for solving (4.13) is

interior-point method with log barrier. This involves absorbing the $\mathbf{G} \succeq 0$ constraint into a term $\frac{1}{t}\log\det(\mathbf{G})$ and iteratively solving

$$\max_{\mathbf{G},v} \gamma\tilde{g}(\mathbf{\Sigma}) + \log p(\mathcal{X}|\mathbf{\Sigma}) - \lambda\mathrm{tr}(\mathbf{G}) + \frac{1}{t}\log\det(\mathbf{G}) \tag{A.12}$$

for an increasing sequence of $t > 0$. As t goes to infinity, its solution converges to that of (4.13). Without loss of generality, let us start this iterative procedure with an initial point $(\mathbf{G}_0, v_0) = (\mathbf{I}_M, 2)$. Obviously $\mathbf{G}_0$ has the same eigenvectors as $\mathbf{\Sigma}_{\mathrm{SAM}}$. Let us denote the objective of (A.12) by $\mathcal{L}_t(\mathbf{G}, v)$. Then we have

$$\begin{aligned}\nabla_{\mathbf{G}}\mathcal{L}_t(\mathbf{G}, v) \;=\; & -\frac{\gamma N}{2}\left(\mathbf{c}\mathbf{c}^{\mathrm{T}} - \frac{1}{2}\mathbf{c}\mathbf{c}^{\mathrm{T}}(v\mathbf{I}-\mathbf{G})\mathbf{\Sigma}_{\mathrm{SAM}} - \frac{1}{2}\mathbf{\Sigma}_{\mathrm{SAM}}(v\mathbf{I}-\mathbf{G})\mathbf{c}\mathbf{c}^{\mathrm{T}}\right) \\ & + \frac{N}{2}\left(\mathbf{\Sigma}_{\mathrm{SAM}} - (v\mathbf{I}-\mathbf{G})^{-1}\right) - \lambda\mathbf{I} + \frac{1}{t}\mathbf{G}^{-1}.\end{aligned}$$

Recall that, if two symmetric matrices $\mathbf{A}$ and $\mathbf{B}$ share the same eigenvectors, then both $\mathbf{A}+\mathbf{B}$ and $\mathbf{AB}$ have the same eigenvectors, too. Since $\mathbf{c}\mathbf{c}^{\mathrm{T}}$, $v_0\mathbf{I}-\mathbf{G}_0$, $\mathbf{\Sigma}_{\mathrm{SAM}}$, $(v_0\mathbf{I}-\mathbf{G}_0)^{-1}$, and $\mathbf{G}_0^{-1}$ are all symmetric and have the same eigenvectors, we know $\nabla_{\mathbf{G}}\mathcal{L}_t|_{(\mathbf{G}_0,v_0)}$ also has the same eigenvectors. Therefore, if we update parameter $\mathbf{G}$ by this gradient, the next $\mathbf{G}$ we arrive at will also have the same eigenvectors. By induction, such eigen-structure is invariant over each iteration. In other words, when we solve (A.12) by gradient ascent method, the resultant $\mathbf{G}$ will still have the same eigenvectors as $\mathbf{\Sigma}_{\mathrm{SAM}}$.

Since this argument is independent of the value of t, as t goes to infinity, such result still holds. Therefore, $\mathbf{G}_\gamma$ has the same eigenvectors as $\mathbf{\Sigma}_{\mathrm{SAM}}$, and so does $\mathbf{\Sigma}_{\mathrm{PEO}}^{\lambda,\epsilon}$.

□

Proposition 4.1. *If* $\mathbf{c} \in \{\mathbf{a}_1, \ldots, \mathbf{a}_M\}$*, for any* $\lambda > 0$ *such that* $\mathcal{G}\left(\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda}\right) \neq \sup_{\mathbf{\Sigma}} \mathcal{G}\left(\mathbf{\Sigma}\right)$*, we have*

$$\mathcal{G}\left(\mathbf{\Sigma}_{\mathrm{PEO}}^{\lambda,\epsilon}\right) > \mathcal{G}\left(\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda}\right), \quad \forall\epsilon > 0.$$

Furthermore, the gap of this inequality monotonically increases with ϵ*.*

Proof. By (4.13), let us rewrite $\mathbf{\Sigma}^{\lambda,\epsilon}_{\text{PEO}}$ as $\mathbf{\Sigma}_\gamma$, where γ is a scalar monotonically increasing with ϵ, and $\gamma = 0$ when $\epsilon = 0$. Since $\mathbf{\Sigma}_{\text{SAM}} = \mathbf{\Sigma}_* = \mathbf{ALA}^{\text{T}}$, by (4.11) we have $\mathcal{F}_\lambda(\mathbf{\Sigma}_*) = \mathbf{\Sigma}^\lambda_{\text{UTM}}$ and therefore $\mathbf{\Sigma}^\lambda_{\text{UTM}} = \mathbf{AWA}^{\text{T}}$, where $\mathbf{W} \in \mathbb{D}^M_+$ is the soft-thresholded version of $\mathbf{L}$. Furthermore, by Lemma A.1 we know $\mathbf{\Sigma}_\gamma$ also has the same eigenvectors $\mathbf{A}$. Let us denote $\mathbf{\Sigma}_\gamma$ by $\mathbf{AH}_\gamma\mathbf{A}^{\text{T}}$, where $\mathbf{H}_\gamma = \text{diag}(h_1(\gamma), \ldots, h_M(\gamma))$, and define $\mathbf{W} = \text{diag}(w_1, \ldots, w_M)$. Obviously $\mathbf{H}_0 = \mathbf{W}$.

Suppose $\mathbf{c} = \mathbf{a}_{i_0}$. By Theorem 4.1, we know

$$\mathcal{F}_\lambda(\mathbf{\Sigma}_{\text{SAM}} + \gamma\mathbf{C} \otimes \mathbf{D}) = \mathbf{\Sigma}_\gamma,$$

where $\mathbf{C} = \mathbf{cc}^{\text{T}} = \mathbf{a}_{i_0}\mathbf{a}^{\text{T}}_{i_0}$ and $\mathbf{D} = \mathbf{\Sigma}^{-1}_\gamma\mathbf{\Sigma}_{\text{SAM}} - \mathbf{I}$. Since $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{\Sigma}_{\text{SAM}}$ all have the same eigenvectors $\mathbf{A}$, we can remove it and write

$$\mathcal{F}_\lambda\left(\mathbf{L} + \gamma\left(\frac{\ell_{i_0}}{h_{i_0}(\gamma)} - 1\right)\mathbf{e}_{i_0}\mathbf{e}^{\text{T}}_{i_0}\right) = \mathbf{H}_\gamma.$$

Let $f_\gamma(h)$ be the i_0-th diagonal entry of matrix $\mathcal{F}_\lambda\left(\mathbf{L} + \gamma\left(\frac{\ell_{i_0}}{h} - 1\right)\mathbf{e}_{i_0}\mathbf{e}^{\text{T}}_{i_0}\right)$. Then $h_{i_0}(\gamma)$ is a solution to the equation $f_\gamma(h) = h$. Also note that $f_\gamma(h)$ is continuous and monotonically decreases with h.

Now consider three cases:

1. $w_{i_0} < \ell_{i_0}$: We have

$$f_\gamma(\ell_{i_0}) = w_{i_0} < \ell_{i_0}$$

and

$$\begin{aligned} f_\gamma(w_{i_0}) &= \text{the } i_0\text{-th diagonal entry of matrix } \mathcal{F}_\lambda\left(\mathbf{L} + \gamma\left(\frac{\ell_{i_0}}{w_{i_0}} - 1\right)\mathbf{e}_{i_0}\mathbf{e}^{\text{T}}_{i_0}\right) \\ &> \text{the } i_0\text{-th diagonal entry of matrix } \mathcal{F}_\lambda(\mathbf{L}) = w_{i_0}. \end{aligned}$$

This is equivalent to $f_\gamma(w_{i_0}) - w_{i_0} > 0$ and $f_\gamma(\ell_{i_0}) - \ell_{i_0} < 0$. Thus, by Intermediate Value Theorem, we know there exists $h' \in (w_{i_0}, \ell_{i_0})$ that satisfies

$$f_\gamma(h') = h'.$$

Furthermore, by the monotonicity of $f_\gamma(h)$, we know the solution to $f_\gamma(h) = h$ is unique, and therefore $h' = h_{i_0}(\gamma)$.

But for all $h \in (w_{i_0}, \ell_{i_0})$, we have $\left(\frac{\ell_{i_0}}{h} - 1\right) > 0$, which implies

$$f_{\gamma_1}(h) > f_{\gamma_2}(h), \quad \forall \gamma_1 > \gamma_2.$$

Therefore, the root of equation $f_\gamma(h) = h$ monotonically increases with γ. This implies that $h_{i_0}(\gamma)$ monotonically increases from w_{i_0} towards ℓ_{i_0}, as γ increases from 0 towards ∞.

Recall that

$$\frac{1}{2}(\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}})^{-1}\mathbf{c} = \frac{\mathbf{a}_{i_0}}{2w_{i_0}}$$

and therefore

$$\begin{aligned} \mathcal{G}\left(\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}\right) &= \mathbf{c}^{\mathrm{T}}\left(\frac{\mathbf{a}_{i_0}}{2w_{i_0}}\right) - \left(\frac{\mathbf{a}_{i_0}}{2w_{i_0}}\right)^{\mathrm{T}} \mathbf{\Sigma}_* \left(\frac{\mathbf{a}_{i_0}}{2w_{i_0}}\right) \\ &= \frac{1}{2w_{i_0}} - \frac{\ell_{i_0}}{4w_{i_0}^2} = \frac{1}{4\ell_{i_0}} - \ell_{i_0}\left(\frac{1}{2w_{i_0}} - \frac{1}{2\ell_{i_0}}\right)^2. \end{aligned}$$

Similarly,

$$\mathcal{G}\left(\mathbf{\Sigma}_\gamma\right) = \frac{1}{4\ell_{i_0}} - \ell_{i_0}\left(\frac{1}{2h_{i_0}(\gamma)} - \frac{1}{2\ell_{i_0}}\right)^2$$

and thus

$$\mathcal{G}\left(\mathbf{\Sigma}_\gamma\right) - \mathcal{G}\left(\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}\right) = \ell_{i_0}\left(\left(\frac{1}{2w_{i_0}} - \frac{1}{2\ell_{i_0}}\right)^2 - \left(\frac{1}{2h_{i_0}(\gamma)} - \frac{1}{2\ell_{i_0}}\right)^2\right),$$

which is positive and monotonically increases as γ increases from 0 towards ∞ and $h_{i_0}(\gamma)$ increases from w_{i_0} towards ℓ_{i_0}.

2. $w_{i_0} = \ell_{i_0}$: In this case, $\mathcal{G}\left(\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}\right) = \mathcal{G}\left(\mathbf{\Sigma}_*\right) = \sup_{\mathbf{\Sigma}} \mathcal{G}\left(\mathbf{\Sigma}\right)$, which is precluded from the assumption.

3. $w_{i_0} > \ell_{i_0}$: Similarly to case 1, we can see that $h_{i_0}(\gamma)$ monotonically decreases from w_{i_0} towards ℓ_{i_0} as γ increases from 0 towards ∞, and therefore the desired

results follow.

□

Proposition 4.2. *Fixing N, K, and σ_*^2, consider a sequence of covariance matrices $\mathbf{\Sigma}_*^{(M)}$ and objective vectors $\mathbf{c}^{(M)}$, indexed by the dimension M, that satisfy $\ell_i^{(M)} \in [\frac{2\lambda}{N} - \delta_i, \frac{2\lambda}{N} + \delta_i]$ for $i = 1, 2, \ldots, K$ and constants $\delta_1, \ldots, \delta_K$. If $\mathbf{c}^{(M)} \in \left\{\mathbf{a}_1^{(M)}, \mathbf{a}_2^{(M)}, \ldots, \mathbf{a}_K^{(M)}\right\}$, we have*

$$\lim_{\epsilon \to \infty} \mathcal{G}(\mathbf{\Sigma}_{\text{PEO}}^{\lambda,\epsilon}) - \mathcal{G}(\mathbf{\Sigma}_{\text{UTM}}^{\lambda}) = \Omega(M).$$

Proof. Suppose $\mathbf{c}^{(M)} = \mathbf{a}_{i_M}^{(M)}$, $i_M \leq K$. By the derivation in Proposition 4.1, we have $\lim_{\epsilon\to\infty} \mathcal{G}(\mathbf{\Sigma}_{\text{PEO}}^{\lambda,\epsilon}) = \lim_{\gamma\to\infty} \mathcal{G}(\mathbf{\Sigma}_{\gamma}) = \frac{1}{4\ell_{i_M}^{(M)}}$, whereas $\mathcal{G}(\mathbf{\Sigma}_{\text{UTM}}^{\lambda})$ only depends on the i_M-th eigenvalue of $\mathbf{\Sigma}_{\text{UTM}}^{\lambda}$, for which we now derive an upper bound.

Let $\hat{\sigma}_{(M)}^2$ be the smallest eigenvalue of $\mathbf{\Sigma}_{\text{UTM}}^{\lambda}$. Since

$$\begin{aligned}
\hat{\sigma}_{(M)}^2 &\leq \frac{1}{M}\text{tr}(\mathbf{\Sigma}_*^{(M)}) = \frac{1}{M}\left(\sum_{i=1}^{K} \ell_i^{(M)} + (M-K)\sigma_*^2\right) \\
&\leq \frac{1}{M}\left(\frac{2K\lambda}{N} + \sum_{i=1}^{K} \delta_i + (M-K)\sigma_*^2\right) \\
&= \frac{1}{M}\left(\frac{2KM\sigma_*^2}{N} + M\sigma_*^2 + \text{constant}\right),
\end{aligned}$$

there exists M_0 such that $\hat{\sigma}_{(M)}^2 \leq \left(\frac{2K}{N} + 1\right)\sigma_*^2 + 1$ for all $M > M_0$. Now let the i_M-th eigenvalue of $\mathbf{\Sigma}_{\text{UTM}}^{\lambda}$ be h_M. By (4.11), we have

$$h_M = \max\left\{\ell_{i_M}^{(M)} - \frac{2\lambda}{N}, \hat{\sigma}_{(M)}^2\right\} \leq \max\left\{\delta_{i_M}, \hat{\sigma}_{(M)}^2\right\}.$$

Letting $h' = \max\left\{\delta_1, \ldots, \delta_K, \left(\frac{2K}{N} + 1\right)\sigma_*^2 + 1\right\}$, we further have $h_M \leq h', \forall M > M_0$. Since h' is a constant and $\ell_1^{(M)}, \ldots, \ell_K^{(M)} = \Omega(M)$, there exists an integer $M_1 > M_0$

such that $\ell_i^{(M)} > h' > h_M$ for all $i = 1, \ldots, K$ and $M > M_1$. Therefore,

$$\begin{aligned}\mathcal{G}(\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda}) &= \frac{1}{4\ell_{i_M}^{(M)}} - \ell_{i_M}^{(M)} \left(\frac{1}{2h_M} - \frac{1}{2\ell_{i_M}^{(M)}}\right)^2 \\ &\leq \frac{1}{4\ell_{i_M}^{(M)}} - \ell_{i_M}^{(M)} \left(\frac{1}{2h'} - \frac{1}{2\ell_{i_M}^{(M)}}\right)^2, \forall M > M_1\end{aligned}$$

and

$$\lim_{\epsilon\to\infty} \mathcal{G}(\mathbf{\Sigma}_{\mathrm{PEO}}^{\lambda,\epsilon}) - \mathcal{G}(\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda}) \geq \ell_{i_M}^{(M)} \left(\frac{1}{2h'} - \frac{1}{2\ell_{i_M}^{(M)}}\right)^2, \forall M > M_1.$$

The desired result then follows from the fact $\ell_{i_M}^{(M)} = \Omega(M)$. □

Proposition 4.3. *Fixing N and σ_*^2, consider a sequence of covariance matrices $\mathbf{\Sigma}_*^{(M)}$ and objective vectors $\mathbf{c}^{(M)}$, indexed by the dimension M, that satisfy $K^{(M)} = \lceil \alpha M \rceil$, where $\alpha \in (0, 1)$ is a constant, and $\ell_i^{(M)} > \frac{2\lambda}{N} + \sigma_*^2$ for $i = 1, 2, \ldots, K^{(M)}$. If $\mathbf{c}^{(M)} \perp \operatorname{span}\left\{\mathbf{a}_1^{(M)}, \mathbf{a}_2^{(M)}, \ldots, \mathbf{a}_{K^{(M)}}^{(M)}\right\}$, we have*

$$\lim_{\epsilon\to\infty} \frac{\mathcal{G}(\mathbf{\Sigma}_{\mathrm{PEO}}^{\lambda,\epsilon})}{\mathcal{G}(\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda})} = \Omega(M).$$

Proof. Since

$$\mathbf{\Sigma}_*^{(M)} = \mathbf{A}\mathbf{L}\mathbf{A}^{\mathrm{T}} = \sum_{i=1}^{K^{(M)}} (\ell_i^{(M)} - \sigma_*^2)\mathbf{a}_i^{(M)}\mathbf{a}_i^{(M)\mathrm{T}} + \sigma_*^2\mathbf{I}$$

and

$$\mathbf{c}^{(M)} \perp \operatorname{span}\left\{\mathbf{a}_1^{(M)}, \mathbf{a}_2^{(M)}, \ldots, \mathbf{a}_K^{(M)}\right\},$$

we know $\mathbf{c}^{(M)}$ is an eigenvector of $\mathbf{\Sigma}_*^{(M)}$ with eigenvalue σ_*^2. Without loss of generality, let $\mathbf{c}^{(M)} = \mathbf{a}_{i_M}^{(M)}$, $i_M > K$. By the derivation in Proposition 4.1, we have $\lim_{\epsilon\to\infty} \mathcal{G}(\mathbf{\Sigma}_{\mathrm{PEO}}^{\lambda,\epsilon}) = \lim_{\gamma\to\infty} \mathcal{G}(\mathbf{\Sigma}_{\gamma}) = \frac{1}{4\sigma_*^2}$, whereas $\mathcal{G}(\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda})$ only depends on the i_M-th eigenvalue of $\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda}$, for which we now derive a lower bound.

Since $\mathbf{\Sigma}_*^{(M)}$ has $K^{(M)}$ outstanding eigenvalues and the remaining $M - K^{(M)}$ ones equal to σ_*^2, the matrix $\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda} = \mathcal{F}_{\lambda}(\mathbf{\Sigma}_*^{(M)})$ can have at most $K^{(M)}$ outstanding eigenvalues. Let the number of outstanding eigenvalues of $\mathbf{\Sigma}_{\mathrm{UTM}}^{\lambda}$ be $\hat{K}^{(M)}$. Since

$i_M \in \{K^{(M)}+1, K^{(M)}+2, \ldots, M\}$, we know the i_M-th eigenvalue of $\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}$ will be the smallest eigenvalue of $\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}$, and by the trace-preservation property of $\mathcal{F}_\lambda$, the smallest eigenvalue of $\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}$ will be

$$\begin{aligned}
\hat{\sigma}^2_{(M)} &= \frac{1}{M-\hat{K}^{(M)}}\left(\frac{2\hat{K}^{(M)}\lambda}{N} + \sum_{i=\hat{K}^{(M)}+1}^{K^{(M)}} \ell_i^{(M)} + (M-K^{(M)})\sigma_*^2\right) \\
&> \frac{1}{M-\hat{K}^{(M)}}\left(\frac{2\hat{K}^{(M)}\lambda}{N} + \left(K^{(M)}-\hat{K}^{(M)}\right)\left(\frac{2\lambda}{N}+\sigma_*^2\right) + (M-K^{(M)})\sigma_*^2\right) \\
&= \frac{1}{M-\hat{K}^{(M)}}\left(\frac{2\hat{K}^{(M)}\lambda}{N} + \left(K^{(M)}-\hat{K}^{(M)}\right)\frac{2\lambda}{N}\right) + \sigma_*^2 \\
&= \frac{1}{M-\hat{K}^{(M)}}\left(\frac{2K^{(M)}\lambda}{N}\right) + \sigma_*^2 = \frac{1}{M-\lceil \alpha M\rceil}\left(\frac{2\lceil\alpha M\rceil M\sigma_*^2}{N}\right) + \sigma_*^2 \\
&\geq \frac{1}{M-\alpha M}\left(\frac{2\alpha M^2\sigma_*^2}{N}\right) + \sigma_*^2 = \frac{2\alpha}{N(1-\alpha)}\cdot M\sigma_*^2 + \sigma_*^2.
\end{aligned}$$

Therefore, there exists a constant $\beta > 0$ such that

$$\hat{\sigma}^2_{(M)} > (\beta M + 1)\sigma_*^2. \tag{A.13}$$

Recall that

$$\mathcal{G}(\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}) = \frac{1}{2\hat{\sigma}^2_{(M)}} - \frac{\sigma_*^2}{4\hat{\sigma}^4_{(M)}} = \frac{1}{2\hat{\sigma}^2_{(M)}}\left(1 - \frac{\sigma_*^2}{2\hat{\sigma}^2_{(M)}}\right).$$

Plugging (A.13) into this we have

$$0 < \mathcal{G}(\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}}) < \frac{1}{M}\cdot\frac{1}{2\beta\sigma_*^2},$$

and therefore

$$\lim_{\epsilon\to\infty}\frac{\mathcal{G}(\mathbf{\Sigma}^{\lambda,\epsilon}_{\mathrm{PEO}})}{\mathcal{G}(\mathbf{\Sigma}^{\lambda}_{\mathrm{UTM}})} = \Omega(M).$$

□

Appendix B

Implementation Details

B.1 Empirical Optimization for Time-Series Regression

The particular local optimization method we use in our computational study is a variation of the Guass-Newton method (Bertsekas, 1999). This suits our problem well because g is quadratic. The particular variation we use deviates from the standard Gauss-Newton method in that each iteration carries out a backtracking line search to determine the size of the step taken toward what would be the next Gauss-Newton iterate. Further, whenever the approximate Hessian is not positive definite, we add to it a small multiple of the identity matrix, in the spirit of the Levenberg-Marquardt method. In our computational study, the backtracking line search starts with a step size of one and halves the step size repeatedly so long as that improves the objective value in (3.3). When a multiple of the identity matrix is added, which is rare, the multiplier is 0.001.

B.2 Choosing PCA Regularization Parameters

For the synthetic data experiment, we select the regularization parameter via the following cross-validation procedure. Let θ be the regularization parameter to be determined and $\mathcal{U}(\mathcal{X}, \theta)$ be the learning algorithm that takes as input $(\mathcal{X}, \theta)$ and returns a covariance matrix estimate. We randomly split $\mathcal{X}$ into a partial training set $\mathcal{X}_{\mathrm{T}}$ and a validation set $\mathcal{X}_{\mathrm{V}}$, whose sizes are roughly 70% and 30% of $\mathcal{X}$, respectively. For each candidate value of θ, $\hat{\mathbf{\Sigma}}^{\theta}_{\mathrm{T}} = \mathcal{U}(\mathcal{X}_{\mathrm{T}}, \theta)$ is computed and the likelihood $p(\mathcal{X}_{\mathrm{V}}|\hat{\mathbf{\Sigma}}^{\theta}_{\mathrm{T}})$ of the validation set $\mathcal{X}_{\mathrm{V}}$ conditioned on the solution $\hat{\mathbf{\Sigma}}^{\theta}_{\mathrm{T}}$ is evaluated. The value of θ that maximizes this likelihood is then selected and fed into $\mathcal{U}(\mathcal{X}, \theta)$ along with the full training set $\mathcal{X}$, resulting in our estimate $\hat{\mathbf{\Sigma}}^{\theta}$. In our implementation, the K for URM/EM are selected from $\{0, 1, \ldots, \ 20\}$, and the λ for UTM/RTM are selected from $\{70, 80, \ldots, 300\}$. These ranges are chosen so that the selected values rarely fall on the extremes.

For the real data experiment, we used the sliding-window validation procedure as described in Algorithm 3. For each algorithm $\mathcal{U} \in \{\mathrm{EM}, \mathrm{RTM}\}$, its regularization parameter was selected by

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \sum_{j=0}^{4} \mathcal{V}(\mathcal{U}, \theta, N, 1250 + 20j).$$

In our implementation, the K for EM is selected from $\{12, 13, \ldots, 22\}$, and the λ for RTM is selected from $\{360, 380, \ldots, 540\}$.

Algorithm 3 Validation Procedure $\mathcal{V}$
Input: learning algorithm $\mathcal{U}$, regularization parameter θ, window size N, time point t
Output: log-likelihood of validation set

$$\mathcal{X}_{\mathrm{T}} \leftarrow \left\{ \mathbf{x}_{(\tau)} \,\middle|\, \mathbf{x}_{(\tau)} = \begin{bmatrix} \frac{y_{1,\tau}}{\eta_{1,\tau}} & \ldots & \frac{y_{M,\tau}}{\eta_{M,\tau}} \end{bmatrix}^{\mathrm{T}}, \tau = t - N + 1, \ldots, t \right\} \text{ // training set}$$

$$\mathcal{X}_{\mathrm{V}} \leftarrow \left\{ \mathbf{x}_{(\tau)} \,\middle|\, \mathbf{x}_{(\tau)} = \begin{bmatrix} \frac{y_{1,\tau}}{\eta_{1,t}} & \ldots & \frac{y_{M,\tau}}{\eta_{M,t}} \end{bmatrix}^{\mathrm{T}}, \tau = t + 1, \ldots, t + 20 \right\} \text{ // validation set}$$

$\hat{\Sigma} \leftarrow \mathcal{U}(\mathcal{X}_{\mathrm{T}}, \theta)$
return $\log p(\mathcal{X}_{\mathrm{V}}|\hat{\mathbf{\Sigma}})$

B.3 S&P500 Data Preprocessing for Directed PCA

Define November 2, 2001 as trading day 1 and August 9, 2007 as trading day 1451. After deleting 47 constituent stocks that are not fully defined over this period, we compute for each stock the daily returns as follows:

1. Let $y'_{i,j}$ be the adjusted close price of stock i on day j, $i = 1, \ldots, 453$ and $j = 1, \ldots, 1451$.
2. Compute the raw daily-return of stock i on day j by

$$y''_{i,j} = \frac{y'_{i,j+1}}{y'_{i,j}} - 1, \quad i = 1, \ldots, 453, \quad j = 1, \ldots, 1450.$$

3. Let $\bar{y}$ be the smallest number such that at least 99.5% of all $y''_{i,j}$ are less than or equal to $\bar{y}$. Let $\underline{y}$ be the largest number such that at least 99.5% of all $y''_{i,j}$'s are greater than or equal to $\underline{y}$. Clip all $y''_{i,j}$ by the interval $[\underline{y}, \bar{y}]$, and denote the resulting value by $y_{i,j}$.
4. Let the volatility of stock i on day $j > 50$ be the 50-day rms $\eta_{i,j} = \sqrt{\frac{1}{50}\sum_{t=1}^{50} y^2_{i,j-t}}$.

Bibliography

P. Abbeel and A. Y. Ng. Learning first-order Markov models for control. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1–8. MIT Press, Cambridge, MA, 2004.

A. A. Amini and M. J. Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. *Ann. Statist.*, 37:2877–2921, 2009.

J.-Y. Audibert. Aggregated estimators and empirical complexity for least square regression. *Annales de l'Institut Henri Poincare Probability and Statistics*, 40(6): 685–736, 2004.

P. L. Bartlett and S. Mendelson. Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334, 2006.

A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23:769–805, 1998.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

D. P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, 2005.

D. Bertsimas and A. Thiele. Robust and data-driven optimization: Modern decision-making under uncertainty. In *Tutorials on Operations Research*. INFORMS, 2006.

O. Besbes, R. Philips, and A. Zeevi. Testing the validity of a demand model: An operations perspective. *Manufacturing & Service Operations Management*, 12(1): 162–183, 2010.

G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley, 2008.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

F. Bunea, A. B. Tsybakov, and M. H. Wegkamp. Aggregation for Gaussian regression. *The Annals of Statistics*, 35(4):1674–1697, 2007.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *CoRR*, 2009.

L. Y. Chua, J. Shanthikumar, and Z. M. Shen. Solving operational statistics via a Bayesian analysis. *Operations Research Letters*, 36:110–116, 2008.

A. D'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3): 434, 2004.

E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 2008.

I. Domowitz and H. White. Misspecified models with dependent observations. *Journal of Econometrics*, 20:35–58, 1982.

D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28:1–38, February 2003.

C. W. J. Granger. Prediction with a generalized cost of error function. *Operation Research*, 20(2):199–207, 1969.

H. H. Harman. *Modern Factor Analysis*. University of Chicago Press, third edition, 1976.

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

I. M. Johnstone and A. Y. Lu. Sparse Principal Components Analysis. *J. Amer. Statist. Assoc*, 2007.

I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

Y.-H. Kao and B. Van Roy. Directed time series regression for control. In *Stanford Technical Report*. 2010.

Y.-H. Kao and B. Van Roy. Estimating a factor model via regularized PCA. 2011.

Y.-H. Kao, B. Van Roy, and X. Yan. Directed regression. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 889–897. 2009.

K. Kim and N. Timm. *Univariate and Multivariate General Linear Models: Theory and Applications with SAS*. Chapman & Hall/CRC, 2006.

M. Landry, S. A. Campbell, K. Morris, and C. O. Aguilar. Dynamics of an inverted pendulum with delayed feedback control. *Journal of Applied Dynamical Systems*, 4(2):333–351, 2005.

L. H. Liyanage and J. G. Shanthikumar. A practical inventory control policy using operational statistics. *Operations Research Letters*, 33:341–348, 2005.

L. Ljung. *System Identification: Theory for the User (2nd Edition)*. Prentice Hall PTR, 1998.

K. E. Muller and P. W. Stewart. *Linear Model Theory: Univariate, Multivariate, and Mixed Models*. Wiley, 2006.

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, Cambridge, MA, 2001.

G. Pison, P. J. Rousseeuw, P. Filzmoser, and C. Croux. Robust factor analysis. *Journal of Multivariate Analysis*, 84(1):145 – 172, 2003.

I. Popescu. Robust mean-covariance solutions for stochastic optimization. *Operations Research*, 55(1):98–112, 2007.

D. B. Rubin and D. T. Thayer. EM algorithm for ML factor analysis. *Psychometrika*, 47(1):69–76, 1982.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society*, 1996.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

H. White. Using least squares to approximate unknown regression functions. *International Economic Review*, 21(1), 1980.

H. Xu, C. Caramanis, and S. Mannor. Principal component analysis with contaminated data: The high dimensional case. In *COLT*, pages 490–502, 2010.

H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15, 2004.