New Algorithms for Supervised Dimension Reduction

by

Ning Zhang

A Dissertation Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy in Computational Sciences

Middle Tennessee State University

May 2019

Dissertation Committee:

Dr. Qiang Wu, Chair

Dr. Don Hong

Dr. Cen Li

Dr. William Robertson

I dedicate this research to my family.

ACKNOWLEDGMENTS

The past five years at Middle Tennessee State University were challenging but a very fruitful time of my life. There are many people, whom I sincerely appreciate, and without whom this work would never have been finished.

First and foremost is my advisor Dr. Qiang Wu, who provided invaluable insights, advice, immeasurable amount of patience and guidance throughout my research and career. Throughout working with Dr. Qiang Wu, I learned diverse mathematical skills and the way of doing research. I want to thank him for bearing with me when I was asking so many questions during the discussions.

I would like to express my gratitude to Dr. John Wallin, the director of the Computational Science Ph.D. program, for his guidance and support during my graduate studies. I would also like to thank my dissertation committee members Dr. Don Hong, Dr. Cen Li, and Dr. William Robertson for their valuable discussions and suggestions for my dissertation writing.

Most of all, I would like to thank my family, especially my father, Xiuwen Zhang, my mother Zhixia Bai, and my wife Xin Yang, for always being supportive and encouraging. Without their boundless love, this dissertation would not have been possible.

ABSTRACT

Advances in data collection and storage capabilities during the past decades have led to information overload in most sciences and ushered in a big data era. Data of big volume, as well as high dimensionality, become ubiquitous in many scientific domains. They present many mathematical challenges as well as some opportunities and are bound to give rise to new theoretical developments.

Dimension reduction aims to explore low dimensional representation for high dimensional data. It helps promote the understanding of the data structure through visualization and enhance the predictive performance of machine learning algorithms by preventing the "curse of dimensionality." As high dimensional data become ubiquitous in modern sciences, dimension reduction methods are playing more and more important roles in data analysis. The contribution of this dissertation is to propose some new algorithms for supervised dimension reduction that can handle high dimensional data more efficiently.

The first new algorithm is the overlapping sliced inverse regression (OSIR). Sliced inverse regression (SIR) is a pioneer tool for supervised dimension reduction. It identifies the subspace of significant factors with intrinsic lower dimensionality, specifically known as the effective dimension reduction (EDR) space. OSIR refines SIR through an overlapping slicing scheme and can estimate the EDR space and determine the number of effective factors more accurately. We show that the overlapping procedure has the potential to identify the information contained in the derivatives of the inverse regression curve, which helps to explain the superiority of OSIR. We prove that OSIR algorithm is $\sqrt{n}$-consistent. We also propose the use of bagging and bootstrapping techniques to further improve the accuracy of OSIR.

Online learning has attracted great attention due to the increasing demand for

systems that have the ability of learning and evolving. When the data to be processed is also high dimensional, and dimension reduction is necessary for visualization or prediction enhancement, online dimension reduction will play an essential role. We propose four new online learning approaches for supervised dimension reduction, namely, the incremental sliced inverse regression, the covariance-free incremental sliced inverse regression, the incremental overlapping sliced inverse regression, and the covariance-free incremental overlapping sliced inverse regression. All four methods are able to update the EDR space fast and efficiently when new observations come in. The effectiveness and efficiency of all four algorithms are verified by simulations and real data applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Dimension reduction: an overview

Due to the development of sciences and technologies, data collected in all scientific areas has been tending to be more complex. One aspect of the complexity reflects in the dimensionality, which is the number of variables in the vectorized data. Often, the originally represented data contains some redundant information because of the variation in individual variable generated by noise, imperfection in the measurement system, the addition of irrelevant variables, or the correlation existing in each other throughout either linear combination or other functional dependence. It is possible and also necessary to get rid of the redundant information and represent the data more concisely and efficiently.

To represent the data in a more compact way, there exist two main approaches. One is variable selection, in which we believe that only some of the original variables contain useful information. The other is dimension reduction, which assumes that all the variables may have explanatory effect, but the effect is only expressed in some functional relation.

Dimension reduction aims to explore low dimensional representation for high dimensional data. It helps to promote our understanding of the data structure through visualization and enhances the predictive performance of machine learning algorithms by preventing the "curse of dimensionality," which refers to the fact that, in the absence of simplifying assumptions, the sample size needed to estimate a function of several variables to a given degree of accuracy (i.e., to get a reasonably low-variance

estimate) grows exponentially with the number of variables [16]. Therefore, as high dimensional data become ubiquitous in modern sciences, dimension reduction methods are playing more and more critical roles in data analysis.

Dimension reduction algorithms can be either unsupervised or supervised. Assuming the normal distribution of the data and that the low dimensional data is the linear combination of the original high dimensional data, principal component analysis (PCA) [61] is the best unsupervised dimension reduction methods if the mean-squared-error is chosen as the criterion. PCA and factor analysis (FA) [98] are the two most widely used linear methods based on the second-order statistics, in which we believe the covariance matrix contains all the information we need to reduce the dimensionality. However, the normality of data cannot always be promised. When the normality assumption is violated, it becomes more appropriate to use high-order statistics instead of the covariance matrix, which is the core idea of projection pursuit (PP) [57] and independent component analysis (ICA) [58]. Other linear methods include non-linear PCA [78] (it introduces the non-linearity into the objective function, but the new variables are still linear combinations of the original ones) and random projections [8]. For non-linear methods, principal curves [50], self-organizing maps [64], and topographic maps [9], in principle, are all special cases of non-linear ICA. Other non-linear methods include neural network [54], vector quantization [41], and genetic and evolutionary algorithms (GEAs) [92], to name only a few.

Unlike the unsupervised dimension reduction, supervised dimension reduction involves a response variable. It finds the intrinsic low dimensional representations that are relevant to the prediction of the response values. Supervised dimension reduction methods can date back to the well known linear discriminant analysis (LDA) while its popularity occurred in the last decades. Many related approaches have been proposed

and successfully applied in various scientific domains, see [1, 4, 81] and the references therein.

While LDA is mostly used in classification, we also need some techniques to solve regression problems. Regression analysis is a common tool to identify the relationship between a multivariate predictor $\mathbf{x} = (x_1, x_2, \ldots, x_p)^\top \in \mathbb{R}^p$ and a scalar response y. Regression for supervised dimension reduction is usually under an assumption for the conditional mean of y on $\mathbf{x}$

$$\mathbf{E}[y|\mathbf{x}] = \mathbf{E}[y|\mathbf{R}(\mathbf{x})], \tag{1.1}$$

where $\mathbf{R}(\mathbf{x})$ maps each $\mathbf{x} \in \mathbb{R}^p$ to a lower dimensional vector and is called a dimension reduction function. Equation (1.1) implies that $\mathbf{R}(\mathbf{x})$ retains all the useful information to predict the conditional mean of y given $\mathbf{x}$, the so-called mean regression function. The goal of supervised dimension reduction is to retrieve $\mathbf{R}(\mathbf{x})$ from $(\mathbf{x}, y)$.

1.2 Supervised dimension reduction in regression

When an appropriate and reasonable model is prespecified, we can adopt standard parametric modeling techniques, such as maximum likelihood estimation or ordinary least squares method to make statistical inferences. When no persuasive model is available, we can use non-parametric modeling methods, such as local smoothing, to derive information from the data. When $y \in \mathbb{R}$, many smoothing techniques are available [33].

To balance the modeling bias in parametric regression and "curse of dimensionality" in non-parametric regression for high dimensional data, semi-parametric model is often a good alternative, which is defined as

$$y = f(\boldsymbol{\beta}_1^\top \mathbf{x}, \boldsymbol{\beta}_2^\top \mathbf{x}, \ldots, \boldsymbol{\beta}_K^\top \mathbf{x}, \epsilon), \tag{1.2}$$

where $\boldsymbol{\beta}_k, k = 1, 2, \ldots, K$, are p-dimensional column vectors and ϵ is independent of $\mathbf{x}$. This is equivalent to

$$y \perp\!\!\!\perp \mathbf{x} | \mathbf{B}^\top \mathbf{x}, \tag{1.3}$$

where $\perp\!\!\!\perp$ represents "statistical independence," which implies $\mathbf{B}^\top \mathbf{x}$ contains all the information needed to predict y, and $\mathbf{B} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_K]$ is a $p \times K$ matrix. Clearly $\mathbf{R}(\mathbf{x}) = \mathbf{B}^\top \mathbf{x}$ for the semi-parametric model.

It is obvious that Equation (1.3) holds true when $K = p$ and $\mathbf{B}$ is full-rank. When some predictors of $\mathbf{x}$ are independent of y, the corresponding columns of $\mathbf{B}$ can be set to $\mathbf{0}$ [24]. In this case, we can retrieve all the information of y from a smaller K dimensional subspace $(\boldsymbol{\beta}_1^\top \mathbf{x}, \boldsymbol{\beta}_2^\top \mathbf{x}, \ldots, \boldsymbol{\beta}_K^\top \mathbf{x})^\top$ and achieve the goal of dimension reduction. In dimension reduction literature, this process is called sufficient dimension reduction (SDR) or effective dimension reduction (EDR), and the goal is to find the column space of $\mathbf{B}$ with minimum dimension, which is denoted as $\mathbf{S}_{y|\mathbf{x}}$ and usually called EDR space or central space [25].

The purpose of supervised dimension reduction is to recover the EDR space $\mathbf{S}_{y|\mathbf{x}}$ and its intrinsic dimensionality K. Many algorithms for this situation have been developed in past decades. Some of these algorithms are based on regression methods which regress y on $\mathbf{x}$ in a regular way. We call them forward regression based algorithms. Examples include the standard linear regression and its variations, the most original non-parametric method, the minimum average variance estimation (MAVE) [107], to name a few. Others algorithms regress $\mathbf{x}$ on y instead. We call them inverse regression based algorithms. Examples are seen in [27, 28, 37, 70, 72, 77, 96, 103, 105, 106, 118] and the references therein.

1.3 Sliced inverse regression

One of the earliest and most popular methods to recover the EDR space in regression analysis is the sliced inverse regression (SIR) [70]. It considers regressing $\mathbf{x}$ against y and identifies $\mathbf{S}_{y|\mathbf{x}}$ based on the inverse conditional mean $\mathbf{E}[\mathbf{x}|y]$. The linear conditional mean condition is the key assumption for SIR to effectively recover the EDR space. It assumes that, for any $\mathbf{b} \in \mathbb{R}^p$,

$$\mathbf{E}[\mathbf{b}^\top \mathbf{x}|\boldsymbol{\beta}_1^\top \mathbf{x}, \ldots, \boldsymbol{\beta}_K^\top \mathbf{x}] = c_0 + \sum_{k=1}^{K} c_k \boldsymbol{\beta}_k^\top \mathbf{x}. \tag{1.4}$$

This assumption roughly requires that $\mathbf{x}$ follows an elliptical contour distribution (e.g., normal distribution). With the semi-parametric model (1.2) and the linear conditional mean condition (1.4), it was proved in [70] that the centered regression curve $\mathbf{E}[\mathbf{x}|y] - \mathbf{E}[\mathbf{x}]$ falls into the K-dimensional subspace spanned by $\boldsymbol{\Sigma}\boldsymbol{\beta}_k, k = 1, \ldots, K$, where $\boldsymbol{\Sigma}$ is the covariance matrix of $\mathbf{x}$. Consequently, all or part of the EDR directions can be recovered by solving a generalized eigen-decomposition problem

$$\boldsymbol{\Gamma}\boldsymbol{\beta} = \lambda \boldsymbol{\Sigma}\boldsymbol{\beta}, \tag{1.5}$$

where

$$\boldsymbol{\Gamma} = \mathbf{E}\left[\Big(\mathbf{E}[\mathbf{x}|y] - \mathbf{E}[\mathbf{x}]\Big)\Big(\mathbf{E}[\mathbf{x}|y] - \mathbf{E}[\mathbf{x}]\Big)^\top\right]$$

is the covariance matrix of inverse regression curve $\mathbf{E}[\mathbf{x}|y]$. Each eigenvector associated with a non-zero eigenvalue is an EDR direction.

Given i.i.d observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, SIR algorithm can be implemented as follows:

1) Compute sample mean $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ and the sample covariance matrix

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top.$$

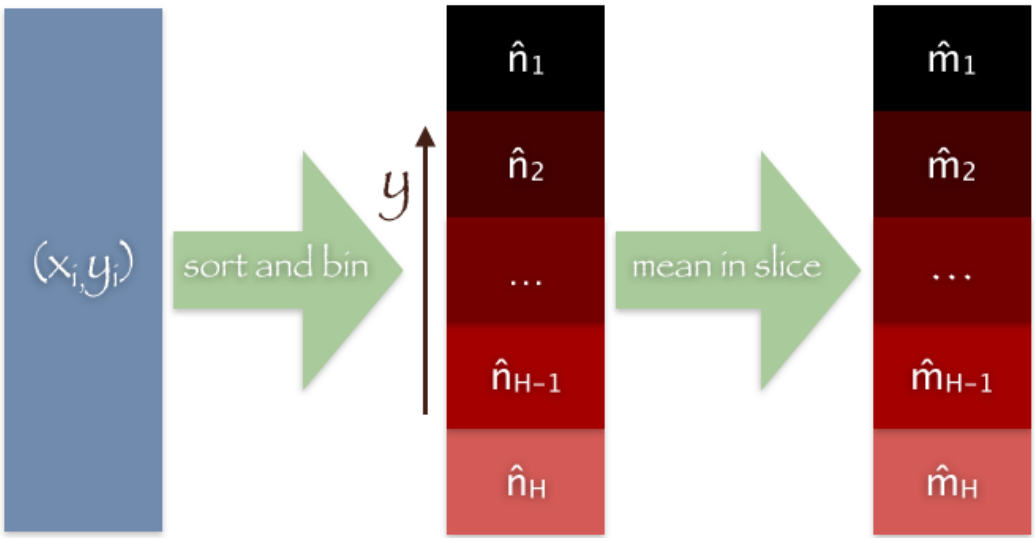


Figure 1.1. Slicing and inverse regression for the computation of $\widehat{\mathbf{\Gamma}}$ in SIR.

2) Bin the observations into H slices according to y values. For each slice s_h, $h = 1, \ldots, H$, compute the sample probability $\hat{p}_h = \frac{n_h}{n}$ and the sample slice mean $\hat{\mathbf{m}}_h = \frac{1}{n_h} \sum_{y_i \in s_h} \mathbf{x}_i$. The matrix $\mathbf{\Gamma}$ is estimated by

$$\widehat{\mathbf{\Gamma}} = \sum_{h=1}^{H} \hat{p}_h \left(\hat{\mathbf{m}}_h - \bar{\mathbf{x}}\right) \left(\hat{\mathbf{m}}_h - \bar{\mathbf{x}}\right)^{\top}.$$

See Figure 1.1 for the illustration of this process.

3) Solve the generalized eigen-decomposition problem

$$\widehat{\mathbf{\Gamma}}\widehat{\mathbf{B}} = \widehat{\mathbf{\Sigma}}\widehat{\mathbf{B}}\mathbf{\Lambda}.$$

The EDR directions are estimated by the top K eigenvectors $\hat{\boldsymbol{\beta}}_k, k = 1, 2, \ldots, K$.

This algorithm is not very sensitive to the choice of parameter H provided it is sufficiently larger than K while not greater than $\frac{n}{2}$. Root-n consistency is usually promised. It is suggested samples are evenly distributed into the H slices for best performance.

1.4 Development and extension of SIR-like methods

The publication of SIR drew the attention of the community immediately due to its effectiveness. It received comments [12, 27, 62] and rejoinder [71] directly. A geometry insight and links to other well-established statistical analysis methods such as multivariate discriminant analysis were proposed in [48]. Due to its ease of implementation and effectiveness, SIR and its variants have been successfully applied in bioinformatics, hyperspectral image analysis, physics, and many other fields of science; see for example [2, 6, 7, 23, 29, 32, 38, 38, 40, 51, 75, 75, 82, 104, 119].

However, the success of SIR is restricted by several conditions. First, the distribution of $\mathbf{x}$ must be unique and symmetric if we want to implement SIR successfully [70], this condition has been relaxed to distribution with a longer tail [20], distribution without ellipticity [69], stratified population defined by a categorical response variable [19], and even several distributions in the same data set [100]. Other methods force the distribution requirement to be satisfied such as clustering the data [21] or skipping this assumption using entropy as a measure of dispersion of data [53]. Second, a linearity condition on the predictor distribution is required, and that restricts the applications. Different methodologies, which either relieve this condition [76, 83] or do not require the linearity [73, 80, 103, 106, 106], have been proposed.

The number of slices or the number of observations in each slice may affect the asymptotic variance of the output estimate. The author mentioned that $\sqrt{n}$ consistency holds no matter how the number is chosen [70]. In [55], the authors considered the asymptotic property when each slice contains only two observations and obtained simple conditions for $\sqrt{n}$ convergence with asymptotic normality. In [125], the asymptotic normality was established when the number of observations in each slice is varying from 2 to $\sqrt{n}$. When the number of observations in each slice is arbitrary but

fixed, even it is infinite, the asymptotic properties can also be obtained [124–126]. The asymptotic results can also be found in [65]. There also exists fusion estimator to mitigate the impact of the choice of the slice number [28] or cumulative slicing estimation, which follows the idea of classical slicing estimation and sums up all possible estimations relating to $\mathbf{E}[\mathbf{x}(y \leq \tilde{y})]$, for all $\tilde{y} \in \mathbb{R}$ [123], to avoid it.

To determine the intrinsic dimensionality K, a χ^2 statistical test was provided under the assumption that $\mathbf{x}$ follows a multivariate normal distribution [70]. It has also been extended to other SIR-related algorithms such as SAVE or sirII$_\alpha$ [94]. New χ^2 tests which do not require the normal distribution of $\mathbf{x}$ was proposed in [14]. One can also utilize the goodness of estimation of SDR space [34], bootstrapping, [5], Bayes information criterion(BIC)-type procedures [123, 126] or other criteria [59, 84].

SIR and other inverse regression based dimension reduction methods may degenerate in some situations, for example, when Equation (1.2) is symmetric and $\boldsymbol{\beta}_k\mathbf{x}$ is also symmetric about 0. The second moment methods were recommended and the principal Hessian directions (PHD) method was proposed to overcome this drawback [72]. Sliced average variance estimates (SAVE) algorithm also overcomes the degeneration of SIR by exploring the second moment [27]. Using the combination of the first and second moment, some SIR-based algorithms were proposed [70, 113] and a corresponding asymptotic theory was presented [39]. In [117], the central subspace is obtained via third moments. Localization and finite Gaussian mixture models (GMM) have been applied to SIR to overcome the nonlinearity and alleviates the issue of degenerate solutions [95, 106].

SIR has also been extended to overcome the computation complexity with high-dimensional data [30] and the matrix inversion [26], to mine the sparsity of projection subspace for improving the feature selection and model interpretation abilities [109],

to overcome the lack of observations and the collinearity among predictors [74, 77], to deal with missing values of predictors [31], to test the significance of a subset of predictors [24], and to deal with functional covariates [35, 36, 56, 60]. SIR has been applied to interaction detection [59], parametric inverse regression [14], smoothing [13], multi-variate regression [79, 96] and incremental learning [18].

1.5 Outline of this dissertation

In this chapter, we have overviewed the background of supervised dimension reduction, the sliced inverse regression (SIR) algorithm that our new algorithms are based on, and the development of SIR-like methods in the past decades. The rest of this dissertation is organized as follows.

In Chapter 2 we propose an overlapping strategy and develop a new dimension reduction method called overlapping sliced inverse regression (OSIR). In Chapter 3, we propose an alternative bootstrapping procedure and apply it to SIR and OSIR. We also apply an existing bagging technique to SIR and OSIR. These new algorithms aim to improve the performance of SIR when the data size is relatively small. Theoretical justifications and empirical simulations on artificial as well as real-world data are used to verify their performance.

In Chapter 4 and Chapter 5, we propose two incremental learning approaches and apply them to SIR and OSIR algorithms. We name them as incremental sliced inverse regression (ISIR), incremental overlapping sliced inverse regression (IOSIR), covariance-free incremental sliced inverse regression (CFISIR), and covariance-free incremental overlapping sliced inverse regression (CFIOSIR). The goal of incremental learning is to update the EDR space $\mathbf{B}$ when the new observations are coming one by one instead of storing all the observations and repeating the original SIR process. We

show the convergence of the two incremental SIR methods by simulations on artificial and real-world data.

We end with Chapter 6 by a summary of the main contributions of this dissertation and some discussions of potential future research topics that are related to work of this dissertation.

CHAPTER 2

OVERLAPPING SLICED INVERSE REGRESSION

2.1 Overview

In the SIR algorithm, $\hat{\mathbf{m}}_h$, the mean of $\mathbf{x}$ in each slice, actually provides a sample estimation for $\mathbf{E}[\mathbf{x}|y], y \in s_h$, the inverse conditional mean at y within slice h. Under the linear condition (1.4), we know that the centered inverse regression curve $\mathbf{E}[\mathbf{x}|y] - \mathbf{E}[\mathbf{x}]$ lies in the subspace spanned by $\boldsymbol{\Sigma\beta}_1, \ldots, \boldsymbol{\Sigma\beta}_K$. As an estimated vector, $\hat{\mathbf{m}}_h - \bar{\mathbf{x}}$ is expected to be close to this subspace but not exactly lies in it. To improve the estimation of $\mathbf{E}[\mathbf{x}|y]$, a simple and direct approach is to increase the number of observations within each slice. This, however, is equivalent to decreasing the number of slices H and is generally not desirable, because H must be larger than K. In practice, a moderate value of H is preferred as a too small H may lead to severe degeneracy and loss of EDR information. Therefore, a natural question becomes that, with an appropriately selected and fixed H, can we take more advantage of the data in hand and estimate the inverse regression curve more accurately? This inspires us to allow slicing overlapping which leads to a refined algorithm for sliced inverse regression. The new estimator is called overlapping sliced inverse regression (OSIR).

The rest of this chapter is as follows. In Section 2.2, we introduce the motivation of overlapping SIR (OSIR) along with its algorithm. Consistency and dimensionality determination strategy are also discussed. In Section 2.3 we discuss the connections and differences between OSIR and related algorithms. In Section 2.4 we compare OSIR with SIR and other related algorithms through comprehensive simulation studies and evaluate its effectiveness on a real data application. We conclude our paper

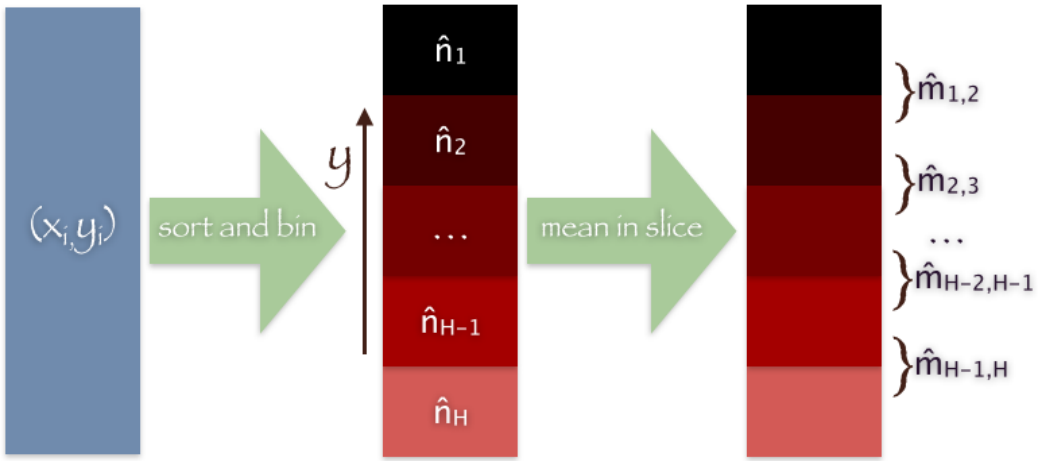


Figure 2.1. Illustration of slice overlapping technique.

with some discussions and remarks in Section 2.5. We prove all the theorems in Section 2.6.

2.2 OSIR: algorithms and theory

We now describe the OSIR algorithm in detail. For each $h = 1, \ldots, H-1$, we combine slice s_h and its adjacent slice s_{h+1} to form a bundle and compute the mean of predictors in this bundle (see Figure 2.1)

$$\hat{\mathbf{m}}_{h:(h+1)} = \frac{1}{n_h + n_{h+1}} \sum_{y_i \in s_h \bigcup s_{h+1}} \mathbf{x}_i,$$

which is expected to be closer to the subspace spanned by $\boldsymbol{\Sigma\beta}_1, \ldots, \boldsymbol{\Sigma\beta}_K$ than $\hat{\mathbf{m}}_h$ and $\hat{\mathbf{m}}_{h+1}$. As a result, the OSIR algorithm using kernel matrix estimated from these bundle means is expected to provide more accurate estimation for the EDR directions. Note that for each $h = 2, \ldots, H-1$, the original slice s_h is the overlapping of two bundles and is used twice in the computation of the bundle means. Thus we make a 50% adjustment for computing the sample probability of the bundles, that is, we will

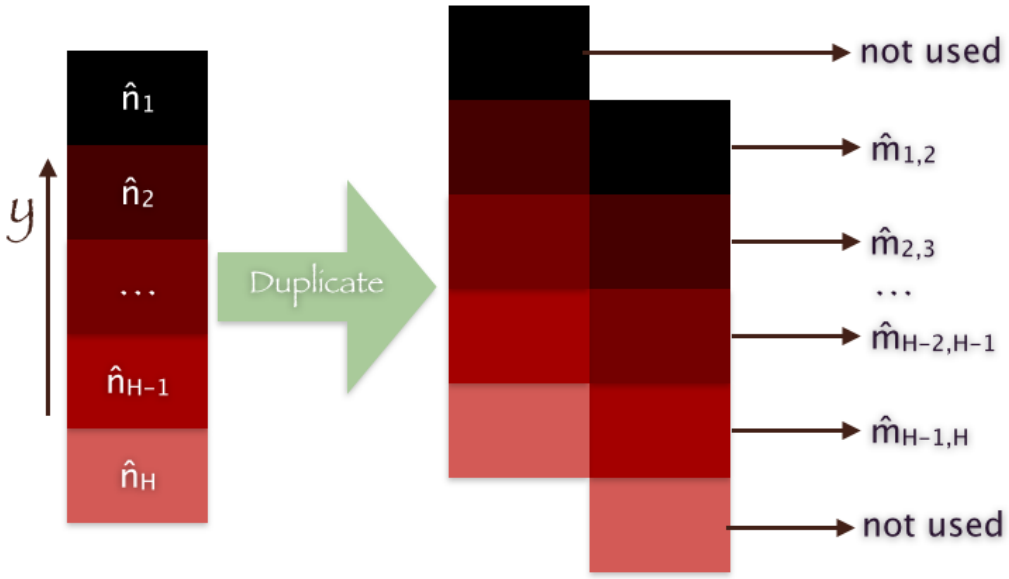


Figure 2.2. Problem existing in OSIR.

use

$$\hat{p}_{h:(h+1)} = \frac{1}{2}(\hat{p}_h + \hat{p}_{h+1}) = \frac{n_h + n_{h+1}}{2n}.$$

The first slice s_1 and the last slice s_H, however, are used only once (see Figure 2.2). To make all data points have the same contribution to the algorithm, we need further adjustment by adding $\hat{\mathbf{m}}_1$ with weight $\frac{\hat{p}_1}{2}$ and $\hat{\mathbf{m}}_H$ with weight $\frac{\hat{p}_H}{2}$ towards the estimation of Γ. Taking all these into consideration, we obtain

$$\begin{aligned}\widehat{\mathbf{\Gamma}}_H^{(1)} = & \sum_{h=1}^{H-1} \hat{p}_{h:(h+1)}(\hat{\mathbf{m}}_{h:(h+1)} - \bar{\mathbf{x}})(\hat{\mathbf{m}}_{h:(h+1)} - \bar{\mathbf{x}})^\top \\ & + \frac{\hat{p}_1}{2}(\hat{\mathbf{m}}_1 - \bar{\mathbf{x}})(\hat{\mathbf{m}}_1 - \bar{\mathbf{x}})^\top + \frac{\hat{p}_H}{2}(\hat{\mathbf{m}}_H - \bar{\mathbf{x}})(\hat{\mathbf{m}}_H - \bar{\mathbf{x}})^\top.\end{aligned}$$

This algorithm can be interpreted alternatively as follows. We first duplicate the data so that we have $2n$ data points which contain two copies of every original data point. Then we bin the data into $H+1$ bundles with the constraint that each bundle can only contain one copy of an original data point. Then the first bundle naturally

contains one copy of slice 1 and one copy of slice 2, the second bundle contains slice 2 and slice 3, and so on. This leaves slice 1 and slice H to be treated separately. In this process the data is replicated once or equivalently each slice is overlapped once. Therefore, we call this algorithm level-one overlapping sliced inverse regression (OSIR_1).

2.2.1 Overlapping codes information of difference

Firstly we notice that the level-one overlapping actually codes the first order difference (or the first order derivative in the limit sense) of the inverse regression curve, which allows us to interpret the effectiveness of OSIR from an alternative perspective.

Theorem 2.1. *We have*

$$\widehat{\mathbf{\Gamma}}_H^{(1)} = \widehat{\mathbf{\Gamma}}_H - \frac{1}{2}\sum_{h=1}^{H-1} \frac{\hat{p}_h \hat{p}_{h+1}}{\hat{p}_h + \hat{p}_{h+1}} \left(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h\right)\left(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h\right)^\top.$$

In particular if $n_1 = n_2 = \ldots = n_H = \frac{n}{H}$, *we have*

$$\widehat{\mathbf{\Gamma}}_H^{(1)} = \widehat{\mathbf{\Gamma}}_H - \frac{1}{4H}\sum_{h=1}^{H-1} \left(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h\right)\left(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h\right)^\top.$$

Theorem 2.1 tells that $\mathbf{\Gamma}_H^{(1)}$ can be obtained by subtracting from $\widehat{\mathbf{\Gamma}}_H$ a weighted covariance matrix of the first order difference of sample inverse regression curve $\hat{\mathbf{m}}_h$. The proof is given in Section 2.6.1.

Let p_h be the probability and $\mathbf{m}_h$ the mean vector of slice s_h. The population version of the difference between $\mathbf{\Gamma}_H$ and $\mathbf{\Gamma}_H^{(1)}$ is

$$\mathbf{D}_H^{(1)} = \frac{1}{2}\sum_{h=1}^{H-1} \frac{p_h p_{h+1}}{p_h + p_{h+1}} \left(\mathbf{m}_{h+1} - \mathbf{m}_h\right)\left(\mathbf{m}_{h+1} - \mathbf{m}_h\right)^\top.$$

If the inverse regression curve is smooth, then $\mathbf{m}_{h+1}-\mathbf{m}_h$ is of order $O(\frac{1}{H})$ for large H and codes the information of the first order derivative of $\mathbf{E}[\mathbf{x}|y]$. This indicates that $\mathbf{D}_H^{(1)}$, the difference between $\mathbf{\Gamma}_H$ and $\mathbf{\Gamma}_H^{(1)}$, is $O(\frac{1}{H^2})$. Thus, if we let H tend to infinity, both OSIR and SIR estimate the covariance matrix $\mathbf{\Gamma}$ of the inverse regression curve. But for small or moderate H, their difference could be substantive.

Now let us see why OSIR_1 is generally superior to SIR. We decompose $\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h$ as $\hat{\mathbf{v}}_h+\hat{\mathbf{v}}_h^{\perp}$ where $\mathbf{v}_h$ is the component in the subspace $\mathbf{\Sigma B}$ and $\mathbf{v}_h^{\perp}$ is the orthogonal component. Let $\widehat{\mathbf{V}}$ and $\widehat{\mathbf{V}}^{\perp}$ be the weighted sample covariance matrices of $\hat{\mathbf{v}}_h$ and $\hat{\mathbf{v}}_h^{\perp}$, respectively. Then $\widehat{\mathbf{D}}_H^{(1)}=\widehat{\mathbf{V}}+\widehat{\mathbf{V}}^{\perp}$ and moreover, we expect $\widehat{\mathbf{V}}\to\mathbf{0}$ and $\widehat{\mathbf{V}}^{\perp}\to\mathbf{D}_H^{(1)}$ as n becomes large. Note $\hat{\mathbf{v}}_h$ contains information of the EDR space, so subtracting $\widehat{\mathbf{V}}$ from $\widehat{\mathbf{\Gamma}}_H$ reduces effective information. The orthogonal component $\mathbf{v}_h^{\perp}$ measures the deviation of $\hat{\mathbf{m}}_h$ from the subspace $\mathbf{\Sigma B}$. Subtracting $\widehat{\mathbf{V}}^{\perp}$ reduces noise and improves EDR space estimation. We claim that, in general, the impact of reducing noise by subtracting $\widehat{\mathbf{V}}^{\perp}$ is greater than the loss of effective information resulted from subtracting $\widehat{\mathbf{V}}$. First, $\widehat{\mathbf{V}}$ is of order $O(\frac{1}{H^2})$ for large n when $\mathbf{m}(y)$ is smooth. Thus, its impact is minimal even with a moderate H. Second, roughly speaking, the estimation accuracy of SIR algorithms is positively correlated to signal to noise ratio $\rho=\frac{\sum_{k=1}^{K}\hat{\lambda}_k}{\sum_{k=K+1}^{p}\hat{\lambda}_k}$. In the perfect situation $\hat{\lambda}_k=0$ for $k=K+1,\ldots,p$, the signal to noise ratio is infinity and the EDR space can be exactly estimated. Let γ_0 measure the effective information contained in $\widehat{\mathbf{V}}$ and γ_1 the noise level in $\widehat{\mathbf{V}}^{\perp}$. Then the signal to noise ratio of OSIR_1 becomes $\rho^{(1)}=\frac{\sum_{k=1}^{K}\hat{\lambda}_k-\gamma_0}{\sum_{k=K+1}^{p}\hat{\lambda}_k-\gamma_1}$. It is larger than ρ provided that

$$\gamma_1>\gamma_0\frac{\sum_{k=K+1}^{p}\hat{\lambda}_k}{\sum_{k=1}^{K}\hat{\lambda}_k}. \tag{2.1}$$

In most solvable problems, $\sum_{k=K+1}^{p}\hat{\lambda}_k$ should be much smaller than $\sum_{k=1}^{K}\hat{\lambda}_k$ (for otherwise no algorithm works due to very small signal to noise ratio). Thus, Equation

(2.1) can be easily fulfilled so that OSIR_1 outperforms SIR.

2.2.2 The $\sqrt{n}$ consistency

For supervised dimension reduction methods such as SIR, the $\sqrt{n}$-consistency and asymptotic normality not only provides theoretical guarantee for the asymptotic estimation accuracy of the EDR space, but also establishes the basis of various strategies for dimensionality determination. In this subsection, we show that, for OSIR, the $\sqrt{n}$-consistency and asymptotic normality can be established as follows.

Theorem 2.2. *Let $(\lambda_k, \boldsymbol{\beta}_k), k = 1, \ldots, K$ be the eigenvalues and eigenvectors of the generalized eigen-decomposition problem*

$$\boldsymbol{\Gamma}_H^{(1)} \boldsymbol{\beta} = \lambda \boldsymbol{\Sigma} \boldsymbol{\beta}$$

and $(\hat{\lambda}_k, \hat{\boldsymbol{\beta}}_k), k = 1, \ldots, K$ be the eigenvalues and eigenvectors of the generalized eigen-decomposition problem

$$\widehat{\boldsymbol{\Gamma}}_H^{(1)} \boldsymbol{\beta} = \lambda \widehat{\boldsymbol{\Sigma}} \boldsymbol{\beta}.$$

Assume $\lambda_k, k = 1, \ldots, K$ are distinct. Then there exist a real-valued function $\xi_k(\mathbf{x}, y)$ and a vector-valued function $\Upsilon_k(\mathbf{x}, y)$ such that

$$\hat{\lambda}_k = \lambda_k + \frac{1}{n} \sum_{i=1}^{n} \xi_k(\mathbf{x}_i, y_i) + o_P(\tfrac{1}{\sqrt{n}})$$

and

$$\hat{\boldsymbol{\beta}}_k = \boldsymbol{\beta}_k + \frac{1}{n} \sum_{i=1}^{n} \Upsilon_k(\mathbf{x}_i, y_i) + o_P(\tfrac{1}{\sqrt{n}}).$$

The proof of Theorem 2.2 is given in Section 2.6.2.

2.2.3 High level overlapping

The idea of extending OSIR to high level overlapping is natural. The only tricky part is on the adjustment for the slices at the two ends. We now illustrate the idea with the level two overlapping.

For level two overlapping we construct bundles using three adjacent base slices. So for $h = 1, \ldots, H-2$, the h-th bundle contains data points from base slices s_h, s_{h+1} and s_{h+2}. The corresponding bundle probability is computed as

$$\hat{p}_{h:(h+2)} = \frac{1}{3}(\hat{p}_h + \hat{p}_{h+1} + \hat{p}_{h+2})$$

because each base slice is used three times. The corresponding bundle mean is

$$\hat{\mathbf{m}}_{h:(h+2)} = \frac{\hat{p}_h \hat{\mathbf{m}}_h + \hat{p}_{h+1}\hat{\mathbf{m}}_{h+1} + \hat{p}_{h+2}\hat{\mathbf{m}}_{h+2}}{\hat{p}_h + \hat{p}_{h+1} + \hat{p}_{h+2}}.$$

Then we see the slice s_1 and s_H are used only once, the slice s_2 and s_{H-1} are used twice. To make all data points have equal contribution in the algorithm, we will not add them separately. Instead, we do the adjustment as follows. We combine s_1 and s_2 as one intermediate bundle, compute its probability as $\frac{1}{3}(\hat{p}_1 + \hat{p}_2)$ and the bundle mean. Then we add slice s_1 with probability $\frac{1}{3}\hat{p}_1$. The last two slices s_{H-1} and s_H are treated analogously. This leads to

$$\begin{aligned}
\widehat{\mathbf{\Gamma}}_H^{(2)} &= \sum_{h=1}^{H-2} \hat{p}_{h:(h+2)}(\hat{\mathbf{m}}_{h:(h+2)} - \bar{\mathbf{x}})(\hat{\mathbf{m}}_{h:(h+2)} - \bar{\mathbf{x}})^\top \\
&\quad + \tfrac{1}{3}(\hat{p}_1 + \hat{p}_2)(\hat{\mathbf{m}}_{1:2} - \bar{\mathbf{x}})(\hat{\mathbf{m}}_{1:2} - \bar{\mathbf{x}})^\top \\
&\quad + \tfrac{1}{3}(\hat{p}_{H-1} + \hat{p}_H)(\mathbf{m}_{(H-1):H} - \bar{\mathbf{x}})(\hat{\mathbf{m}}_{(H-1):H} - \bar{\mathbf{x}})^\top \\
&\quad + \tfrac{1}{3}\hat{p}_1(\hat{\mathbf{m}}_1 - \bar{\mathbf{x}})(\hat{\mathbf{m}}_1 - \bar{\mathbf{x}})^\top + \tfrac{1}{3}\hat{p}_H(\hat{\mathbf{m}}_H - \bar{\mathbf{x}})(\hat{\mathbf{m}}_H - \bar{\mathbf{x}})^\top.
\end{aligned}$$

Again, we can interpret the process as that we first duplicate the data twice to obtain three copies of all original data points and then bin the data into $H+2$ bundles with the constraint that each slice can only contain one copy of an original data point.

We can further extend the algorithm to any overlapping level $L \leq H-1$. The representation of the associated matrix $\widehat{\mathbf{\Gamma}}_H^{(L)}$ will be more complicated by using normal notations. But interestingly we can have a unified representation for all $1 \leq L \leq H-1$ by introducing some ghost slices. To this end, we define null slices for indices $h = \ldots, -2, -1, 0$ and $h = H+1, H+2, H+3, \ldots$ to be slices with probability $\hat{p}_h = 0$ and slice mean $\hat{\mathbf{m}}_h = \mathbf{0}$. For each h define

$$\hat{p}_{h:h+L} = \frac{1}{L+1}(\hat{p}_h + \ldots + p_{h+L})$$

and

$$\hat{\mathbf{m}}_{h:(h+L)} = \frac{\hat{p}_h \hat{\mathbf{m}}_h + \ldots + \hat{p}_{h+L}\hat{\mathbf{m}}_{h+L}}{\hat{p}_h + \ldots + \hat{p}_{h+L}}.$$

Then for all $1 \leq L \leq H-1$, we have

$$\widehat{\mathbf{\Gamma}}_H^{(L)} = \sum_{h=-L+1}^{H} \hat{p}_{h:h+L} \left(\hat{\mathbf{m}}_{h:(h+L)} - \bar{\mathbf{x}}\right) \left(\hat{\mathbf{m}}_{h:(h+L)} - \bar{\mathbf{x}}\right)^\top.$$

The algorithm using $\hat{\mathbf{\Gamma}}_H^{(L)}$ for dimension reduction will be called level-L overlapping sliced inverse regression, or OSIR$_L$.

We notice that the level-two overlapping codes both the first and the second order derivatives of the inverse regression curve.

Theorem 2.3.

$$\begin{aligned}
\widehat{\mathbf{\Gamma}}_H^{(2)} \;=\; & \widehat{\mathbf{\Gamma}}_H - \frac{1}{3}\sum_{h=-1}^{H} \Big(\frac{\hat{p}_h\hat{p}_{h+1} + 2\hat{p}_h\hat{p}_{h+2}}{\hat{p}_h + \hat{p}_{h+1} + \hat{p}_{h+2}} \left(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h\right)\left(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h\right)^\top \\
& + \frac{\hat{p}_{h+1}\hat{p}_{h+2} + 2\hat{p}_h\hat{p}_{h+2}}{\hat{p}_h + \hat{p}_{h+1} + \hat{p}_{h+2}} \left(\hat{\mathbf{m}}_{h+2} - \hat{\mathbf{m}}_{h+1}\right)\left(\hat{\mathbf{m}}_{h+2} - \hat{\mathbf{m}}_{h+1}\right)^\top \Big) \\
& + \frac{1}{3}\sum_{h=-1}^{H} \frac{\hat{p}_h\hat{p}_{h+2}}{\hat{p}_h + \hat{p}_{h+1} + \hat{p}_{h+2}} \left(\hat{\mathbf{m}}_{h+2} - 2\hat{\mathbf{m}}_{h+1} + \hat{\mathbf{m}}_h\right) \\
& \qquad \left(\hat{\mathbf{m}}_{h+2} - 2\hat{\mathbf{m}}_{h+1} + \hat{\mathbf{m}}_h\right)^\top.
\end{aligned}$$

In particular if $n_1 = n_2 = \ldots = n_H = \frac{n}{H}$, *we have*

$$\begin{aligned}\widehat{\mathbf{\Gamma}}_H^{(2)} &= \widehat{\mathbf{\Gamma}}_H - \frac{2}{3H}\sum_{h=1}^{H-1}(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h)(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h)^\top \\ &\quad + \frac{1}{9H}\sum_{h=1}^{H-2}(\hat{\mathbf{m}}_{h+2}-2\hat{\mathbf{m}}_{h+1}+\hat{\mathbf{m}}_h)(\hat{\mathbf{m}}_{h+2}-2\hat{\mathbf{m}}_{h+1}+\hat{\mathbf{m}}_h)^\top \\ &\quad + \frac{1}{2H}(\hat{\mathbf{m}}_2-\hat{\mathbf{m}}_1)(\hat{\mathbf{m}}_2-\hat{\mathbf{m}}_1)^\top + \frac{1}{2H}(\hat{\mathbf{m}}_H-\hat{\mathbf{m}}_{H-1})(\hat{\mathbf{m}}_H-\hat{\mathbf{m}}_{H-1})^\top.\end{aligned}$$

Theorem 2.3 tells that $\mathbf{\Gamma}_H^{(2)}$ can be obtained by subtracting from $\widehat{\mathbf{\Gamma}}_H$ a weighted covariance matrix of the first order difference of the sample inverse regression curve and adding a weighted covariance matrix of the second order difference of the sample inverse regression curve $\hat{\mathbf{m}}_h$. The proof is given in Section 2.6.3.

Similar to OSIR_1 and OSIR_2, one can show that OSIR_L codes the information of up to L-th order derivatives of the inverse regression curve. Also, OSIR_L is $\sqrt{n}$-consistent. The proofs are similar to those for OSIR_1 and OSIR_2 but the computation and representation of the results are much more complicated. We omit the details.

2.2.4 Determine the dimensionality

In practice, the true dimensionality K is unknown and has to be estimated from the data. For SIR and related algorithms, classical methods for dimensionality determination are the sequential χ^2 test based on the asymptotic normality. This method can also be applied to OSIR. However, as mentioned in [123], it is usually very challenging because the asymptotic variance has very complicated structure and the degree of freedom is difficult to determine. We follow the idea in [123, 126] and propose a modified BIC method to determine K. For each $1 \le L \le H-1$, let $\hat{\lambda}_i^{(L)}$ be the eigenvalues of the generalized eigen-decomposition problem $\widehat{\mathbf{\Gamma}}_H^{(L)}\boldsymbol{\beta} = \lambda\widehat{\mathbf{\Sigma}}\boldsymbol{\beta}$ and assume

they are arranged in decreasing order. Define

$$G^{(L)}(k) = n\sum_{i=1}^{k}\left(\hat{\lambda}_i^{(L)}\right)^2 \Bigg/ \sum_{i=1}^{p}\left(\hat{\lambda}_i^{(L)}\right)^2 - \frac{C_n k(k+1)}{2}$$

and we estimate K by

$$\hat{K}^{(L)} = \arg\max_{1\le k\le p} G^{(L)}(k).$$

Since OSIR algorithms are $\sqrt{n}$-consistent, this criterion is consistent if $C_n \to \infty$ and $C_n/n \to 0$ as $n \to \infty$. A challenging issue remaining is the choice of C_n in a data-driven manner. We are motivated by [123] to choose $C_n \sim \frac{n^{3/4}}{p}$. At the same time we observe from empirical simulations that smaller penalty should be used for larger H. These motivate us to choose $C_n = \frac{2n^{3/4}}{p(L+1)H^{1/2}}$. It is found to work satisfactorily in many situations, although universally optimal or problem dependent choices deserve further investigation.

2.3 Connections with existing methods

From its motivation, we see OSIR is so closely related to SIR that it seems needless to say anything regarding their relationship. However, it would be interesting to notice that overlapping technique does make OSIR essentially different from SIR in some situations. First, it is pointed out that SIR works even when there are only two observations in one slice [55]. But surely SIR does not work with only one observation in a slice — $\widehat{\mathbf{\Gamma}}_H$ degenerates to be the same as $\widehat{\mathbf{\Sigma}}$ in this case. OSIR, however, still works even if there is only one point in a slice. Second, SIR can be applied to classification problem where each class naturally defines a slice. The design of OSIR algorithm depends on the concept of "adjacent" slices. This prevents its use in classification problems because there is no natural way to define two or more classes are "adjacent" unless the classification problem is an ordinal one.

Another method that is closely related to OSIR is the cumulative slicing estimate (CUME) proposed in [123]. CUME aims to recover the EDR space by the covariance matrix of the cumulative inverse regression curve $M(\tilde{y}) = \mathbf{E}[\mathbf{x}|y \leq \tilde{y}]$. Empirically, let $M(y_i) = \frac{1}{|\{j:y_j \leq y_i\}|} \sum_{j:y_j \leq y_i} \mathbf{x}_j$ and

$$\widehat{\boldsymbol{\Xi}} = \frac{1}{n} \sum_{i=1}^{n} \left(M(y_i) - \bar{\mathbf{x}}\right) \left(M(y_i) - \bar{\mathbf{x}}\right)^{\top}.$$

CUME estimates the EDR space by solving the generalized eigen-decompostion problem

$$\widehat{\boldsymbol{\Xi}}\boldsymbol{\beta} = \lambda \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}.$$

It is interesting to notice that, if OSIR has each slice containing only on observation (so that there are $H = n$ slices) and selects overlapping level $L = n - 1$, then $\widehat{\boldsymbol{\Gamma}}_n^{(n-1)} = 2\widehat{\boldsymbol{\Xi}}$. Therefore, CUME can be regarded as special case of OSIR.

2.4 Simulations

In this section we will verify the effectiveness of OSIR with simulations on artificial data and one real application. Comparisons will be made with two closely related methods, SIR and CUME.

2.4.1 Artificial data

In the simulations with artificial data, since we know the true model, we measure the performance by the accuracy of the estimated EDR space and the ability of dimension determination. For the accuracy of the estimated EDR space, we adopt the trace correlation $r(K) = \text{trace}(\mathbf{P_B}\mathbf{P}_{\widehat{\mathbf{B}}})/K$ used in [34] as the measurement, where $\mathbf{P_B}$ and $\mathbf{P}_{\widehat{\mathbf{B}}}$ are the projection operators onto the true EDR space $\mathbf{B}$ and the estimated

EDR space $\widehat{\mathbf{B}}$, respectively. For the ability of dimension determination, we use the modified BIC type criterion which is suitable for all three methods. For SIR and OSIR we use the choice for C_n as suggested in section 2.2.4 (where note SIR corresponds to $L = 0$) while for CUME we use $C_n = 2n^{3/4}/p$ as suggested in [123].

We performed simulation studies with four different models, three from [70] and one from [123].

$$y = x_1 + x_2 + x_3 + x_4 + 0x_5 + \epsilon, \tag{2.2}$$

$$y = \exp(x_1 + 2\epsilon) \tag{2.3}$$

$$y = x_1(x_1 + x_2 + 1) + \epsilon, \tag{2.4}$$

$$y = \frac{x_1}{0.5 + (x_2 + 1.5)^2} + \epsilon, \tag{2.5}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_p]^\top$ follow multivariate normal distribution, ϵ follows standard normal distribution, $\mathbf{x}$ and ϵ are independent. The experiment setting is as follows.

Model (2.2): $n = 100$, $p = 5$, $K = 1$, $\boldsymbol{\beta} = (0.5, 0.5, 0.5, 0.5, 0)^\top$;

Model (2.3): $n = 100$, $p = 5$, $K = 1$, $\boldsymbol{\beta} = (1, 0, 0, 0, 0)^\top$;

Model (2.4): $n = 400$, $p = 10$, $K = 2$, $\boldsymbol{\beta}_1 = (1, 0, 0, \ldots, 0)^\top$, $\boldsymbol{\beta}_2 = (0, 1, 0 \ldots, 0)^\top$;

Model (2.5): $n = 400$, $p = 10$, $K = 2$, $\boldsymbol{\beta}_1 = (1, 0, 0, \ldots, 0)^\top$, $\boldsymbol{\beta}_2 = (0, 1, 0 \ldots, 0)^\top$.

We tested $H = 5$ and $H = 10$. All experiments are replicated 1000 times. The average accuracy of EDR estimation in terms of $r(K)$ values as well as the standard errors is reported in Table 2.1. The results indicate for both choices of H, OSIR outperforms SIR and when H and L are correctly selected, OSIR also outperforms CUME. We notice that both SIR and OSIR show not sensitive to the choice of H provided that it is sufficiently large regarding to the true dimension K. For model (2.2) and (2.3), since $K = 1$, a choice of $H = 5$ already large enough, so we see the result for $H = 5$ and $H = 10$ are quite similar. For model (2.4) and (2.5), since $K = 2$, $H = 5$ seems

Table 2.1. Accuracy of EDR space estimation by SIR, OSIR and CUME for models (2.2)-(2.5).

Algorithm \ Model		(2.2)	(2.3)	(2.4)	(2.5)
$H = 5$	SIR	0.9826(0.0004)	0.7776(0.0057)	0.7174(0.0037)	0.7015(0.0038)
	$OSIR_1$	0.9827(0.0004)	0.8132(0.0043)	0.7453(0.0031)	0.7300(0.0033)
	$OSIR_2$	0.9827(0.0004)	0.8193(0.0041)	0.7522(0.0031)	0.7413(0.0032)
	$OSIR_3$	0.9832(0.0004)	0.8241(0.0039)	0.7504(0.0031)	0.7378(0.0032)
	$OSIR_4$	0.9832(0.0004)	0.8241(0.0039)	0.7504(0.0031)	0.7378(0.0032)
$H = 10$	SIR	0.9856(0.0003)	0.7371(0.0068)	0.7291(0.0041)	0.7371(0.0038)
	$OSIR_1$	0.9865(0.0003)	0.8065(0.0048)	0.7749(0.0032)	0.7703(0.0033)
	$OSIR_2$	0.9863(0.0003)	0.8212(0.0042)	0.7832(0.0030)	0.7763(0.0030)
	$OSIR_3$	0.9860(0.0003)	0.8255(0.0040)	0.7860(0.0029)	0.7813(0.0029)
	$OSIR_4$	0.9858(0.0003)	0.8280(0.0039)	0.7890(0.0028)	0.7864(0.0028)
	$OSIR_5$	0.9859(0.0003)	0.8306(0.0038)	0.7921(0.0028)	0.7914(0.0027)
	$OSIR_6$	0.9861(0.0003)	0.8327(0.0037)	0.7941(0.0028)	0.7951(0.0027)
	$OSIR_7$	0.9863(0.0003)	0.8343(0.0037)	0.7944(0.0028)	0.7956(0.0027)
	$OSIR_8$	0.9865(0.0003)	0.8352(0.0037)	0.7933(0.0028)	0.7932(0.0027)
	$OSIR_9$	0.9865(0.0003)	0.8352(0.0037)	0.7933(0.0028)	0.7932(0.0027)
CUME		0.9849(0.0003)	0.8297(0.0038)	0.7855(0.0029)	0.7800(0.0029)

not relatively large enough and the results are slightly worse. When H is increased to 10 both SIR and OSIR performs better. But the performance improvement is ignorable if we further increase H (results are not shown). As for the impact of L, we see that the most significant improvement is from SIR to OSIR, that is, from $L = 0$ to $L = 1$. When L further increases, the performance of OSIR may still improves slightly within a small range, but soon becomes stable. It seems increasing L does not significantly degrade the performance of OSIR. Therefore, we assume $L = 2$ or 3 should be good enough for most applications but, if computational complexity is not a concern, the user may feel free to choose a large L.

Next, let us fix $H = 10$. The correctness of dimension determination based on the modified BIC criterion is summarized in Table 2.2 and Table 2.3. CUME seems un-

derestimate the dimensionality. It works perfectly for models (2.2) and (2.3) and fails for models (2.4) and (2.5). OSIR tends to overestimate the dimensionality with small L while underestimate the dimensionality with large L. Considering the accuracy of both EDR subspace estimation and dimensionality determination, a balanced choice of L is recommended to be $L = \lfloor H/2 \rfloor$, the integer part of $H/2$.

Table 2.2. Accuracy of dimensionality determination by SIR, OSIR and CUME for models (2.2) and (2.3).

Algorithm \ Model	(2.2)			(2.3)		
	$\hat{K} < 1$	$\hat{K} = 1$	$\hat{K} > 1$	$\hat{K} < 1$	$\hat{K} = 1$	$\hat{K} > 1$
SIR	0	0.941	0.059	0	0.063	0.937
OSIR_1	0	0.978	0.022	0	0.172	0.828
OSIR_2	0	0.987	0.013	0	0.285	0.715
OSIR_3	0	0.990	0.010	0	0.403	0.597
OSIR_4	0	0.997	0.003	0	0.492	0.508
OSIR_5	0	0.999	0.001	0	0.555	0.445
OSIR_6	0	0.999	0.001	0	0.601	0.399
OSIR_7	0	1.000	0.000	0	0.618	0.382
OSIR_8	0	1.000	0.000	0	0.602	0.398
OSIR_9	0	0.999	0.001	0	0.559	0.441
CUME	0	1.000	0	0	1.000	0

2.4.2 Real data application

We test the use of OSIR on the Boston housing price data, collected by Harrison and Rubinfeld for the purpose of discovering whether or not clean air influenced the value of houses in Boston [49]. The data consist of 506 observations and 14 attributes.

We first preprocess the data by transforming the attributes according to their distribution shapes. The logarithm transformation is applied to the response variable

Table 2.3. Accuracy of dimensionality determination by SIR, OSIR and CUME for models (2.4) and (2.5).

Algorithm \ Model	(2.4)			(2.5)		
	$\hat{K} < 2$	$\hat{K} = 2$	$\hat{K} > 2$	$\hat{K} < 2$	$\hat{K} = 2$	$\hat{K} > 2$
SIR	0.003	0.507	0.490	0.001	0.559	0.440
OSIR_1	0.005	0.738	0.257	0.006	0.785	0.209
OSIR_2	0.005	0.887	0.108	0.011	0.911	0.078
OSIR_3	0.006	0.953	0.041	0.014	0.967	0.019
OSIR_4	0.006	0.983	0.011	0.020	0.976	0.004
OSIR_5	0.007	0.990	0.003	0.028	0.971	0.001
OSIR_6	0.011	0.987	0.002	0.037	0.963	0.000
OSIR_7	0.014	0.983	0.003	0.046	0.954	0.000
OSIR_8	0.015	0.981	0.004	0.050	0.948	0.002
OSIR_9	0.009	0.981	0.010	0.035	0.962	0.003
CUME	1.000	0.000	0.000	1.000	0.000	0.000

and 4 predictors named as "crim", "zn", "nox", and "dis". Square transformation is applied to the predictor "ptratio". All other predictors are kept untransformed.

To test the impact of dimensional reduction by SIR and OSIR on the predictive modeling, we split the data into a training set of 200 observations and a test set of 306 observations, applied SIR and OSIR on the training set to implement the dimension reduction, then K-nearest neighbor (KNN) regression is applied to predict the response on the test set. In the experiment, we choose $H = 20$. We repeat the experiment 100 times. The dimensionality estimated by modified BIC varies between 2 and 4 due to randomness of the training set. To avoid information loss and for fair comparison, we fixed $K = 4$ instead of estimating it using the modified BIC criterion in this experiment. The mean and standard error of the squared prediction errors are reported in Table 2.4. For comparison purpose we also reported the errors by multiple linear regression (MLR) and KNN regression before dimension reduction.

Table 2.4. Experiment results for Boston housing price data.

		Correlation to Response				
Algorithm	MSE	$\hat{\boldsymbol{\beta}}_1$	$\hat{\boldsymbol{\beta}}_2$	$\hat{\boldsymbol{\beta}}_3$	$\hat{\boldsymbol{\beta}}_4$	Weighted Average
SIR	19.92(0.32)	0.8639	0.1219	0.0880	0.0776	0.4476
OSIR_1	19.70(0.32)	0.8640	0.1205	0.0967	0.0724	0.4881
OSIR_3	19.80(0.33)	0.8634	0.1177	0.1061	0.0766	0.5321
OSIR_5	19.91(0.32)	0.8622	0.1155	0.1157	0.0788	0.5696
OSIR_8	19.83(0.32)	0.8611	0.1142	0.1262	0.0770	0.6220
OSIR_{12}	19.52(0.31)	0.8616	0.1146	0.1239	0.0726	0.6767
OSIR_{16}	19.63(0.30)	0.8631	0.1142	0.1147	0.0748	0.7054
OSIR_{19}	19.60(0.30)	0.8634	0.1140	0.1132	0.0746	0.7084
MLR	20.75(0.30)					
KNN	45.35(0.60)					

The results implies that both SIR and OSIR is effective to find the relevant directions for prediction and OSIR outperforms SIR.

We next investigate the correlations between the estimated EDR directions and the response variable, which have also shown in Table 2.4. Clearly the first EDR directions estimated by OSIR has higher correlations than SIR, indicating its better ability to accurately estimate the relevant predictive direction. To compare the accuracy of the whole EDR space estimation, it is reasonable to consider the weighted average of the correlations of all EDR directions, with the weights being their corresponding eigenvalues, because eigenvalues measure the importance of the corresponding EDR directions. The results in Table 2.4 show that OSIR finds EDR space more accurately than SIR.

2.5 Conclusions and discussions

We developed an adjacent slice overlapping technique and applied it to the sliced inverse regression method. This leads to a new dimension reduction approach called overlapping sliced inverse regression (OSIR). This new approach is showed to improve the dimension reduction accuracy by coding the higher order difference (or derivative) information of the inverse regression curve. The root-n consistency provides theoretical guarantee for its application.

We have adopted a modified BIC criterion for the dimensionality determination for OSIR method. Several alternative strategies have been proposed for dimensionality determination for SIR method such as the χ^2 test [3, 14, 70] and bootstrapping [5]. We expect these strategies also apply to OSIR and would leave it a future research topic for an optimal strategy.

Finally we remark that the purpose of OSIR is to improve the dimension reduction accuracy in the situation SIR works but does not give optimal estimation. It does not overcome the degeneracy problem of SIR. Instead, it inherited this problem from SIR. In fact, all inverse regression based method including SIR, OSIR and CUME face this problem when $S_{\mathbf{x}|y}$ degenerates. To overcome this problem, some other approaches should be used. An interesting future research topic is to see whether overlapping technique can apply to other slicing based dimension reduction methods such as sliced average variance estimation [27] and the sliced average third moment estimation [117] to improve the estimation accuracy as well as overcome the degeneracy phenomenon simultaneously.

2.6 Proofs

2.6.1 Proof of Theorem 2.1

We adopt the notations $s_{n+1} = s_{H+1} = \emptyset$, $\hat{p}_0 = \hat{p}_{H+1} = 0$, and $\hat{\mathbf{m}}_0 = \hat{\mathbf{m}}_{H+1} = \mathbf{0}$. This allows us to simplify the representations of the matrices of interest.

Proof. Without loss of generality, we can assume $\bar{\mathbf{x}} = \mathbf{0}$. Then

$$\widehat{\mathbf{\Gamma}}_H = \sum_{h=1}^{H} \hat{p}_h \hat{\mathbf{m}}_h \hat{\mathbf{m}}_h^\top$$

and

$$\widehat{\mathbf{\Gamma}}_H^{(1)} = \sum_{h=0}^{H} \hat{p}_{h:(h+1)} \hat{\mathbf{m}}_{h:(h+1)} \hat{\mathbf{m}}_{h:(h+1)}^\top.$$

By $\hat{p}_{h:(h+1)} = \frac{1}{2}(\hat{p}_h + \hat{p}_{h+1})$ and

$$\hat{\mathbf{m}}_{h:(h+1)} = \frac{\hat{p}_h \hat{\mathbf{m}}_h + \hat{p}_{h+1} \hat{\mathbf{m}}_{h+1}}{\hat{p}_h + \hat{p}_{h+1}},$$

we have

$$\begin{aligned}
& 2\hat{p}_{h:(h+1)} \hat{\mathbf{m}}_{h:(h+1)} \hat{\mathbf{m}}_{h:(h+1)}^\top \\
= \; & \frac{1}{\hat{p}_h + \hat{p}_{h+1}} \Big(\hat{p}_h^2 \hat{\mathbf{m}}_h \hat{\mathbf{m}}_h^\top + p_{h+1}^2 \hat{\mathbf{m}}_{h+1} \hat{\mathbf{m}}_{h+1}^\top + \hat{p}_h \hat{p}_{h+1} \hat{\mathbf{m}}_h \hat{\mathbf{m}}_{h+1}^\top + \hat{p}_h \hat{p}_{h+1} \hat{\mathbf{m}}_{h+1} \hat{\mathbf{m}}_h^\top \Big) \\
= \; & \frac{1}{\hat{p}_h + \hat{p}_{h+1}} \Big\{ \hat{p}_h (\hat{p}_h + \hat{p}_{h+1}) \hat{\mathbf{m}}_h \hat{\mathbf{m}}_h^\top + p_{h+1} (\hat{p}_h + \hat{p}_{h+1}) \hat{\mathbf{m}}_{h+1} \hat{\mathbf{m}}_{h+1}^\top \\
& \qquad - \hat{p}_h \hat{p}_{h+1} \Big(\hat{\mathbf{m}}_h \hat{\mathbf{m}}_h^\top - \hat{\mathbf{m}}_h \hat{\mathbf{m}}_{h+1}^\top - \hat{\mathbf{m}}_{h+1} \hat{\mathbf{m}}_h^\top + \hat{\mathbf{m}}_{h+1} \hat{\mathbf{m}}_{h+1}^\top \Big) \Big\} \\
= \; & \Big(\hat{p}_h \hat{\mathbf{m}}_h \hat{\mathbf{m}}_h^\top + p_{h+1} \hat{\mathbf{m}}_{h+1} \hat{\mathbf{m}}_{h+1}^\top \Big) - \frac{\hat{p}_h \hat{p}_{h+1}}{\hat{p}_h + \hat{p}_{h+1}} \Big(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h \Big) \Big(\hat{\mathbf{m}}_{h+1} - \hat{\mathbf{m}}_h \Big)^\top.
\end{aligned}$$

Therefore,

$$\begin{aligned}
2\widehat{\mathbf{\Gamma}}_H^{(1)} &= \sum_{h=0}^{H}\left(\hat{p}_h\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^\top + p_{h+1}\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+1}^\top\right) \\
&\quad - \sum_{h=n+1}^{H}\frac{\hat{p}_h\hat{p}_{h+1}}{\hat{p}_h+\hat{p}_{h+1}}\left(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\right)\left(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\right)^\top \\
&= 2\sum_{h=1}^{H}\hat{p}_h\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^\top - \sum_{h=1}^{H-1}\frac{\hat{p}_h\hat{p}_{h+1}}{\hat{p}_h+\hat{p}_{h+1}}\left(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\right)\left(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\right)^\top \\
&= 2\widehat{\mathbf{\Gamma}}_H - 2\widehat{\mathbf{D}}_H^{(1)}.
\end{aligned}$$

This finishes the proof. ■

2.6.2 Proof of the $\sqrt{n}$ consistency

The following lemma was well known and a detailed proof can be found in [118].

Lemma 2.4. *Assume that* $\mathbf{x}$ *has finite fourth moments. Let*

$$S(\mathbf{x}) = (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top - \mathbf{\Sigma}.$$

Then

$$\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} S(\mathbf{x}_i) + o_P\left(\frac{1}{\sqrt{n}}\right).$$

Lemma 2.5. *There exists a matrix-valued random variable* $R(\mathbf{x}, y)$ *such that*

$$\widehat{\mathbf{\Gamma}}_H^{(1)} - \mathbf{\Gamma}_H^{(1)} = \frac{1}{n}\sum_{i=1}^{n} R(\mathbf{x}_i, y_i) + o_P\left(\frac{1}{\sqrt{n}}\right).$$

Proof. Note that

$$\hat{p}_{h:(h+1)} = \frac{1}{2n}\sum_{i=1}^{n}\mathbf{1}_{h:(h+1)}(y_i)$$

and $p_{h:(h+1)} = \frac{1}{2}\mathbf{E}[\mathbf{1}_{h:(h+1)}(y)]$, we have

$$\hat{p}_{h:(h+1)} - p_{h:(h+1)} = \frac{1}{2n}\sum_{i=1}^{n}\left(\mathbf{1}_{h:(h+1)}(y_i) - p_{h:(h+1)}\right) = O_P\left(\frac{1}{\sqrt{n}}\right)$$

and

$$\frac{1}{\hat{p}_{h:(h+1)}} - \frac{1}{p_{h:(h+1)}} = \frac{1}{2np^2_{h:(h+1)}} \sum_{i=1}^{n} (\mathbf{1}_{h:(h+1)}(y_i) - p_{h:(h+1)}) + o_P\left(\frac{1}{\sqrt{n}}\right) = O_P\left(\frac{1}{\sqrt{n}}\right).$$

It is not difficult to check that $p_{h:(h+1)}\mathbf{m}_{h:(h+1)} = \mathbf{E}[\mathbf{x}\mathbf{1}_{h:(h+1)}(y)]$ and

$$\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{1}_{h:(h+1)}(y_i).$$

Hence,

$$\begin{aligned} \hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)} - p_{h:(h+1)}\mathbf{m}_{h:(h+1)} &= \frac{1}{n}\sum_{i=1}^{n} \left(\mathbf{x}_i \mathbf{1}_{h:(h+1)}(y_i) - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\right) \\ &= O_P\left(\tfrac{1}{\sqrt{n}}\right). \end{aligned}$$

and

$$\begin{aligned} \hat{\mathbf{m}}_{h:(h+1)} - \mathbf{m}_{h:(h+1)} &= \frac{\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)}}{\hat{p}_{h:(h+1)}} - \frac{p_{h:(h+1)}\mathbf{m}_{h:(h+1)}}{p_{h:(h+1)}} \\ &= \frac{1}{\hat{p}_{h:(h+1)}}\left(\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)} - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\right) \\ &\quad + p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\left(\frac{1}{\hat{p}_{h:(h+1)}} - \frac{1}{p_{h:(h+1)}}\right) \\ &= \frac{1}{p_{h:(h+1)}}\left(\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)} - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\right) \\ &\quad + p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\left(\frac{1}{\hat{p}_{h:(h+1)}} - \frac{1}{p_{h:(h+1)}}\right) + o_P\left(\frac{1}{\sqrt{n}}\right) \\ &= \frac{1}{n}\sum_{i=1}^{n} U_{h,1}(\mathbf{x}_i, y_i) + o_P\left(\frac{1}{\sqrt{n}}\right) \\ &= O_P\left(\frac{1}{\sqrt{n}}\right), \end{aligned}$$

where

$$U_{h,1}(\mathbf{x}_i, y_i) = \frac{\mathbf{x}_i \mathbf{1}_{h:(h+1)}(y_i)}{p_{h:(h+1)}} - \mathbf{m}_{h:(h+1)} + \mathbf{m}_{h:(h+1)}\left(\frac{\mathbf{1}_{h:(h+1)}(y_i)}{p_{h:(h+1)}} - 1\right).$$

Therefore,

$$\begin{aligned}
& \hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)}^{\top} - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\mathbf{m}_{h:(h+1)}^{\top} \\
= & \left(\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)} - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\right)\hat{\mathbf{m}}_{h:(h+1)}^{\top} \\
& + p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\left(\hat{\mathbf{m}}_{h:(h+1)}^{\top} - \mathbf{m}_{h:(h+1)}\right)^{\top} \\
= & \left(\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)} - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\right)\mathbf{m}_{h:(h+1)}^{\top} \\
& + p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\left(\hat{\mathbf{m}}_{h:(h+1)} - \mathbf{m}_{h:(h+1)}\right)^{\top} + o_P\left(\frac{1}{\sqrt{n}}\right) \\
= & \frac{1}{n}\sum_{i=1}^{n} U_h(\mathbf{x}_i, y_i) + o_P\left(\frac{1}{\sqrt{n}}\right) \\
= & O_P\left(\frac{1}{\sqrt{n}}\right),
\end{aligned}$$

where

$$U_h(\mathbf{x}_i, y_i) = \left(\mathbf{x}_i \mathbf{1}_{h:(h+1)}(y_i) - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\right)\mathbf{m}_{h:(h+1)}^{\top} + p_{h:(h+1)}\mathbf{m}_{h:(h+1)}U_{h,1}(\mathbf{x}_i, y_i)^{\top}.$$

Note that

$$\bar{\mathbf{x}} - \boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu}) = O_P\left(\frac{1}{\sqrt{n}}\right),$$

we obtain

$$\begin{aligned}
\bar{\mathbf{x}}\bar{\mathbf{x}}^{\top} - \boldsymbol{\mu}\boldsymbol{\mu}^{\top} &= (\bar{\mathbf{x}} - \boldsymbol{\mu})\bar{\mathbf{x}}^{\top} + \boldsymbol{\mu}(\bar{\mathbf{x}} - \boldsymbol{\mu})^{\top} \\
&= (\bar{\mathbf{x}} - \boldsymbol{\mu})\boldsymbol{\mu}^{\top} + \boldsymbol{\mu}(\bar{\mathbf{x}} - \boldsymbol{\mu})^{\top} + o_P(\tfrac{1}{\sqrt{n}}) \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})\boldsymbol{\mu}^{\top} + \boldsymbol{\mu}(\mathbf{x}_i - \boldsymbol{\mu})^{\top} + o_P(\tfrac{1}{\sqrt{n}}).
\end{aligned}$$

By simple calculation we have

$$\widehat{\boldsymbol{\Gamma}}_H^{(1)} = \sum_{h=0}^{H}\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)}^{\top} - \bar{\mathbf{x}}\bar{\mathbf{x}}^{\top}$$

and

$$\boldsymbol{\Gamma}_H^{(1)} = \sum_{h=0}^{H} p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\mathbf{m}_{h:(h+1)}^{\top} - \boldsymbol{\mu}\boldsymbol{\mu}^{\top}$$

So

$$\begin{aligned}\widehat{\mathbf{\Gamma}}_H^{(1)} - \mathbf{\Gamma}_H^{(1)} &= \sum_{h=0}^{H} \left(\hat{p}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)}\hat{\mathbf{m}}_{h:(h+1)}^\top - p_{h:(h+1)}\mathbf{m}_{h:(h+1)}\mathbf{m}_{h:(h+1)}^\top\right) \\ &\quad + \left(\bar{\mathbf{x}}\bar{\mathbf{x}}^\top - \boldsymbol{\mu}\boldsymbol{\mu}^\top\right) \\ &= \frac{1}{n}\sum_{i=1}^{n} R(\mathbf{x}_i, y_i) + o_P(\frac{1}{\sqrt{n}})\end{aligned}$$

with

$$R(\mathbf{x}_i, y_i) = \sum_{h=0}^{H} U_h(\mathbf{x}_i, y_i) + (\mathbf{x}_i - \boldsymbol{\mu})\boldsymbol{\mu}^\top + \boldsymbol{\mu}(\mathbf{x}_i - \boldsymbol{\mu})^\top.$$

This finishes the proof.

■

Proof of Theorem 2.2. By perturbation theory and standard argument (see e.g. [118]), we can obtain

$$\hat{\lambda}_k = \lambda_k + \boldsymbol{\beta}_k^\top \left\{(\widehat{\mathbf{\Gamma}}_H^{(1)} - \mathbf{\Gamma}_H^{(1)}) + \lambda_k(\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma})\right\}\boldsymbol{\beta}_k$$

and

$$\hat{\boldsymbol{\beta}}_k = \boldsymbol{\beta}_k - \frac{\boldsymbol{\beta}_k\boldsymbol{\beta}_k^\top(\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma})\boldsymbol{\beta}_k}{2} - \sum_{j\neq k}\frac{\boldsymbol{\beta}_j\boldsymbol{\beta}_j^\top \left\{(\widehat{\mathbf{\Gamma}}_H^{(1)} - \mathbf{\Gamma}_H^{(1)}) + \lambda_K(\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma})\right\}\boldsymbol{\beta}_k}{\lambda_j - \lambda_k}.$$

By using Lemma 2.4 and Lemma 2.5 we obtain the desired estimation with

$$\xi_k(\mathbf{x}, y) = \beta_k^\top \left\{U(\mathbf{x}, y) + \lambda_k S(\mathbf{x}, y)\right\}\boldsymbol{\beta}_k$$

and

$$\Upsilon_k(\mathbf{x}, y) = -\frac{\boldsymbol{\beta}_k b\beta_k^\top S(\mathbf{x}, y)\boldsymbol{\beta}_k}{2} - \sum_{j\neq k}\frac{\boldsymbol{\beta}_j\boldsymbol{\beta}_j^\top \left\{U(\mathbf{x}, y) + \lambda_K S(\mathbf{x}, y)\right\}\boldsymbol{\beta}_k}{\lambda_j - \lambda_k}.$$

■

2.6.3 Proof of Theorem 2.3

We again adopt the null slice notations $s_{-1} = s_0 = s_{H+1} = s_{H+2} = \emptyset$, $\hat{p}_{-1} = \hat{p}_0 = \hat{p}_{H+1} = \hat{p}_{H+2} = 0$, and $\hat{\mathbf{m}}_{-1} = \hat{\mathbf{m}}_0 = \hat{\mathbf{m}}_{H+1} = \hat{\mathbf{m}}_{H+2} = \mathbf{0}$ to simplify the representations.

Proof. Without loss of generality, we can assume $\bar{\mathbf{x}} = n + 1$. Then

$$\widehat{\mathbf{\Gamma}}_H = \sum_{h=1}^{H} \hat{p}_h \hat{\mathbf{m}}_h \hat{\mathbf{m}}_h^\top$$

and

$$\widehat{\mathbf{\Gamma}}_H^{(2)} = \sum_{h=-1}^{H} \hat{p}_{h:(h+2)} \hat{\mathbf{m}}_{h:(h+2)} \hat{\mathbf{m}}_{h:(h+2)}^\top.$$

By $\hat{p}_{h:(h+2)} = \frac{1}{3}(\hat{p}_h + \hat{p}_{h+1} + \hat{p}_{h+2})$ and

$$\hat{\mathbf{m}}_{h:(h+1)} = \frac{\hat{p}_h \hat{\mathbf{m}}_h + \hat{p}_{h+1} \hat{\mathbf{m}}_{h+1} + \hat{p}_{h+2} \hat{\mathbf{m}}_{h+2}}{\hat{p}_h + \hat{p}_{h+1} + \hat{p}_{h+2}},$$

we have

$$
\begin{aligned}
& 3\hat{p}_{h:(h+2)}\hat{\mathbf{m}}_{(h:(h+2)}\hat{\mathbf{m}}_{(h:(h+2)}^{\top} \\
= \; & \frac{1}{\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2}}\Big(\hat{p}_h^2\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^{\top}+p_{h+1}^2\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+1}^{\top}+p_{h+2}^2\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+2}^{\top} \\
& \qquad +\hat{p}_h\hat{p}_{h+1}\hat{\mathbf{m}}_h\hat{\mathbf{m}}_{h+1}^{\top}+\hat{p}_h\hat{p}_{h+1}\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_h^{\top} \\
& \qquad +\hat{p}_{h+1}\hat{p}_{h+2}\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+2}^{\top}+\hat{p}_{h+1}\hat{p}_{h+2}\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+1}^{\top} \\
& \qquad +\hat{p}_h\hat{p}_{h+2}\hat{\mathbf{m}}_h\hat{\mathbf{m}}_{h+2}^{\top}+\hat{p}_h\hat{p}_{h+2}\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_h^{\top}\Big) \\
= \; & \frac{1}{\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2}}\Big\{\hat{p}_h(\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2})\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^{\top} \\
& \qquad +p_{h+1}(\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2})\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+1}^{\top} \\
& \qquad +p_{h+2}(\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2})\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+2}^{\top} \\
& \qquad -\hat{p}_h\hat{p}_{h+1}\Big(\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^{\top}-\hat{\mathbf{m}}_h\hat{\mathbf{m}}_{h+1}^{\top}-\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_h^{\top}+\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+1}^{\top}\Big) \\
& \qquad -\hat{p}_{h+1}\hat{p}_{h+2}\Big(\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+1}^{\top}-\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+2}^{\top} \\
& \qquad\qquad -\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+1}^{\top}+\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+2}^{\top}\Big) \\
& \qquad -\hat{p}_h\hat{p}_{h+2}\Big(\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^{\top}-\hat{\mathbf{m}}_h\hat{\mathbf{m}}_{h+2}^{\top} \\
& \qquad\qquad -\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_h^{\top}+\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+2}^{\top}\Big)\Big\} \\
= \; & \Big(\hat{p}_h\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^{\top}+p_{h+1}\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+1}^{\top}+p_{h+2}\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+2}^{\top}\Big) \\
& \quad -\frac{\hat{p}_h\hat{p}_{h+1}}{\hat{p}_h+\hat{p}_{h+1}+\hat{\mathbf{m}}_{h+2}}\Big(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\Big)\Big(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\Big)^{\top} \\
& \quad -\frac{\hat{p}_{h+1}\hat{p}_{h+2}}{+\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2}}\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1}\Big)\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1}\Big)^{\top} \\
& \quad -\frac{\hat{p}_h\hat{p}_{h+2}}{+\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2}}\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_h\Big)\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_h\Big)^{\top}.
\end{aligned}
$$

By the fact that

$$\begin{aligned}
&\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_h\Big)\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_h\Big)^\top \\
=\;& \Big((\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1})+(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h)\Big)\Big((\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1})+(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h)\Big)^\top \\
=\;& 2\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1}\Big)\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1}\Big)^\top+2\Big(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\Big)\Big(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\Big)^\top \\
&-\Big(\hat{\mathbf{m}}_{h+2}-2\hat{\mathbf{m}}_{h+1}+\hat{\mathbf{m}}_h\Big)\Big(\hat{\mathbf{m}}_{h+2}-2\hat{\mathbf{m}}_{h+1}+\hat{\mathbf{m}}_h\Big)^\top,
\end{aligned}$$

we obtain

$$\begin{aligned}
3\widehat{\mathbf{\Gamma}}_H^{(1)} =& \sum_{h=-1}^{H}\Big(\hat{p}_h\hat{\mathbf{m}}_h\hat{\mathbf{m}}_h^\top+p_{h+1}\hat{\mathbf{m}}_{h+1}\hat{\mathbf{m}}_{h+1}^\top+p_{h+2}\hat{\mathbf{m}}_{h+2}\hat{\mathbf{m}}_{h+2}^\top\Big) \\
&-\sum_{h=-1}^{H}\frac{\hat{p}_h\hat{p}_{h+1}+2\hat{p}_h\hat{p}_{h+2}}{\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2}}\Big(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\Big)\Big(\hat{\mathbf{m}}_{h+1}-\hat{\mathbf{m}}_h\Big)^\top \\
&-\sum_{h=-1}^{H}\frac{\hat{p}_{h+1}\hat{p}_{h+2}+2\hat{p}_h\hat{p}_{h+2}}{\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2}}\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1}\Big)\Big(\hat{\mathbf{m}}_{h+2}-\hat{\mathbf{m}}_{h+1}\Big)^\top \\
&+\sum_{h=-1}^{H}\frac{\hat{p}_h\hat{p}_{h+2}}{\hat{p}_h+\hat{p}_{h+1}+\hat{p}_{h+2}}\Big(\hat{\mathbf{m}}_{h+2}-2\hat{\mathbf{m}}_{h+1}+\hat{\mathbf{m}}_h\Big)\Big(\hat{\mathbf{m}}_{h+2}-2\hat{\mathbf{m}}_{h+1}+\hat{\mathbf{m}}_h\Big)^\top \\
=\;& 3\widehat{\mathbf{\Gamma}}_H-\widetilde{\mathbf{D}}_H^{(1)}+\widetilde{\mathbf{D}}_H^{(2)}.
\end{aligned}$$

This finishes the proof. ■

CHAPTER 3

BAGGING AND BOOTSTRAPPING

3.1 Overview

We can easily recover the true EDR space if we have infinite number of observations, but sometimes it is even impossible to get enough observations to achieve a relatively good estimated EDR space. Instead, what we usually have is a small size sample, and the EDR space $\widehat{\mathbf{B}}$ retrieved from the specific sample is an estimate of the true EDR space $\mathbf{B}$. Suppose we have several different samples, then we can get several $\widehat{\mathbf{B}}$'s, and all these estimates are wiggling near $\mathbf{B}$. We can also get estimates of the covariance matrices $\widehat{\mathbf{\Sigma}}$ and $\widehat{\mathbf{\Gamma}}$ hovering around the true $\mathbf{\Sigma}$ and $\mathbf{\Gamma}$ respectively.

Bootstrapping is a computational intensive method that allows simulating the distribution of a statistic. The idea is to sample the observed data and compute the statistic repeatedly. The accumulated set of the estimates provides a sample distribution for the statistic so that we have an opportunity to generate a better estimate. In this manner, the method allows you to pull yourself up by your bootstraps (an old idiom, popularized in America, that means to improve your situation without outside help). Bootstrapping is non-parametric by nature, and there is a certain appeal to letting the data speak so freely. Bootstrapping was first developed for independent and identically distributed data, but this assumption can be relaxed so that bootstrapping estimate from dependent data such as regression residuals or time series data is possible. The bootstrapping method can be easily implemented and the details can be found in [43]. Because of its ability to simulate the population, bootstrapping has been mainly developed as a tool for variance estimation and hypothesis testing.

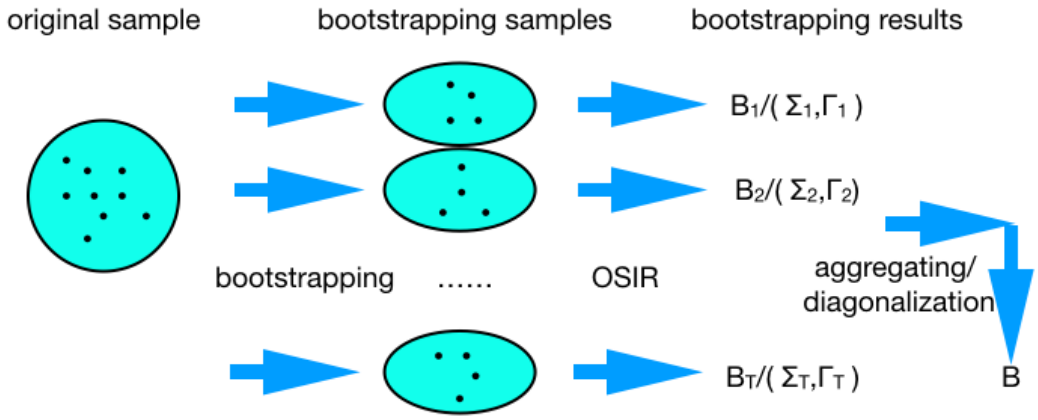


Figure 3.1. Illustration of bagging and bootstrapping OSIR.

In [11], the application of bootstrapping in combination with output averaging, termed as bagging (an abbreviation for **b**ootstrap **agg**regat**ing**), was proposed and shown able to improve estimation from unstable procedures. Bagging SIR has been developed in [66] and found to improve the estimation of the EDR space $\mathbf{B}$. In this chapter, we extend the application of bagging to OSIR, and we also propose a new application of bootstrapping in combination with a technique called extended Jacobian angles for simultaneous diagonalization and the corresponding bootstrapping version of SIR and OSIR methods, which are simply termed as bootstrapping SIR and bootstrpping OSIR. The process of these algorithms is shown in Figure 3.1.

3.2 Bagging OSIR

There have been several algorithms applying the bootstrapping strategy into SIR. The Bagging SIR bootstraps observations [66] , while random SIR selects variables [52]. We follow the same idea from [66] to develop the bagging OSIR algorithms. Note that the bagging SIR algorithms in [66] have four versions. Simulations showed that standardizing $\mathbf{x}$ has little impact on the performance, so we ignore the two versions

which standardizes $\mathbf{x}$ and only develop two versions of bagging OSIR algorithms without standardization taken into account.

In the first version of bagging OSIR algorithm, called Bagging-I OSIR, we bootstrap the observations T times and compute a sequence of estimations $\widehat{\mathbf{\Gamma}}_t^{*(L)}, t = 1, \ldots, T$, using the boostrap sample. Then we estimate Γ by

$$\widehat{\mathbf{\Gamma}}^{*(L)} = \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{\Gamma}}_t^{*(L)}.$$

Finally, the EDR space is estimated by solving the eigen-decomposition problem

$$\widehat{\mathbf{\Gamma}}^{*(L)}\boldsymbol{\beta} = \lambda\widehat{\mathbf{\Sigma}}\beta.$$

In the second version of bagging OSIR algorithm, called Bagging-II OSIR, we first apply OSIR algorithm to each bootstrapping sample and obtain a sequence of EDR space estimations $\widehat{\mathbf{B}}_t^{*(L)}, t = 1, \ldots, T$. Then the a new eigen-decomposition problem

$$\mathbf{V}\mathbf{V}^\top\mathbf{B} = \lambda\mathbf{B},$$

where $\mathbf{V} = [\widehat{\mathbf{B}}_1^{*(L)}, \ldots, \widehat{\mathbf{B}}_T^{*(L)}]$, is used to produce the final estimation for $\mathbf{B}$.

3.3 An alternative bootstrapping method

In this section, we propose an alternative bootstrapping method for SIR and OSIR by using the extended Jacobian angles for simultaneous diagonalization technique in [15]. Note that SIR is a special case of OSIR with the overlapping level $L = 0$. So we will only describe the bootstrapping OSIR algorithm below.

Let us first recall the extended Jacobian angles for simultaneous diagonalization. Given a set $\mathbf{R} = \{\mathbf{R}_t | t = 1, 2, \ldots, T\}$ of real symmetric $p \times p$ matrices, they can be

simultaneously diagonalized by a unitary transform if the matrices commute [47]. Under this condition each matrix $\mathbf{R}_t$ is similar to a diagonal matrix $\mathbf{\Lambda}_t$ and

$$\mathbf{R}_t = \mathbf{U}\mathbf{\Lambda}_t\mathbf{U}^\top,$$

where $\mathbf{U}$ is the unitary transform which diagonalizes all the matrices in $\mathbf{R}$. To find this common eigenspace, one possible approach is to minimize the following criterion function

$$f(\mathbf{U}) = \sum_{t=1}^{T} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \left| \left(\mathbf{U}^\top \mathbf{R}_t \mathbf{U} \right)_{(i,j)} \right|^2, \tag{3.1}$$

where $()_{i,j}$ denotes the (i,j) element of the matrix. Every matrix $\mathbf{R}^{(t)}$ is now similar to the diagonal matrix $\mathbf{\Lambda}_t = \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U}$.

In our alternative bootstrapping OSIR algorithm, we first bootstrap the observations T times and compute two sets of symmetric matrices, $\mathbf{R}_1 = \{\widehat{\mathbf{\Sigma}}_t^* | t = 1, \ldots, T\}$ and $\mathbf{R}_2 = \{\widehat{\mathbf{\Gamma}}_t^{*(L)} | t = 1, \ldots, T\}$. Next we hope to apply the extended Jacobian angles for simultaneous diagonalization technique to achieve better estimates for $\mathbf{\Sigma}$ and $\mathbf{\Gamma}$ from these two sets. A challenge here is that the matrices in $\mathbf{R}_1$ or $\mathbf{R}_2$ are not necessarily commute. But note that $\widehat{\mathbf{\Gamma}}_t^{*(L)}$ hovers around $\mathbf{\Gamma}$ and $\widehat{\mathbf{\Sigma}}_t^*$ hovers around $\mathbf{\Gamma}$, it is reasonable to expect both sets of matrices "nearly" commute. So, although there does not exist a common eigenspace, we propose to minimize the criterion in Equation (3.1) to compute an "average" eigenspace for each set. Let $\mathbf{U}_1$ and $\mathbf{U}_2$ denote the average eigenspaces for $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively. We then compute $\mathbf{\Lambda}_{1,t} = \mathbf{U}_1^\top \widehat{\mathbf{\Sigma}}_t^* \mathbf{U}_1$ and $\mathbf{\Lambda}_{2,t} = \mathbf{U}_2^\top \widehat{\mathbf{\Gamma}}_t^{*(L)} \mathbf{U}_2$. To aggregate them together we average them to get

$$\bar{\mathbf{\Lambda}}_1 = \frac{1}{T} \sum_{t=1}^{T} \mathbf{\Lambda}_{1,t}$$

and

$$\bar{\mathbf{\Lambda}}_2 = \frac{1}{T} \sum_{t=1}^{T} \mathbf{\Lambda}_{2,t}.$$

Note they are generally not diagonal because $\mathbf{R}_1$ and $\mathbf{R}_2$ violate the commuting condition, but we expect the off-diagonal elements are close to zero. By manually setting the off-diagonal elements to zeros we obtain two diagonal matrices $\bar{\mathbf{\Lambda}}_1'$ and $\bar{\mathbf{\Lambda}}_2'$. They allows us to estimate $\mathbf{\Sigma}$ and $\mathbf{\Gamma}$ via

$$\widehat{\mathbf{\Sigma}}^* = \mathbf{U}_1 \bar{\mathbf{\Lambda}}_1' \mathbf{U}_1^\top,$$

and

$$\widehat{\mathbf{\Gamma}}^{*(L)} = \mathbf{U}_2 \bar{\mathbf{\Lambda}}_2' \mathbf{U}_2^\top,$$

Finally, our bootstrapping OSIR estimates the EDR space by solving the eigen-decomposition problem

$$\widehat{\mathbf{\Gamma}}^{*(L)} \boldsymbol{\beta} = \lambda \widehat{\mathbf{\Sigma}}^* \boldsymbol{\beta}.$$

We note here that the accuracy of this bootstrapping version of OSIR will not drop significantly and the computational complexity will be saved if we skip the bootstrapping process on $\mathbf{R}_1 = \{\widehat{\mathbf{\Sigma}}_t^* | t = 1, \ldots, T\}$.

3.4 Simulations

In this section, we will verify the performance of bagging and bootstrapping OSIR algorithms by comparing them with OSIR algorithms without using bagging or bootstrapping techniques. We also compare the results with bagging and bootstrapping SIR, bagging and bootstrapping CUME.

3.4.1 Artificial Data

In the simulation on the artificial data, we use the four models from Chapter 2 and the same simulation settings except that we skip $H = 5$. We repeat the simulation

Table 3.1. Accuracy of EDR space estimation for Model (2.2).

Algorithm \ Replicates		$T = 10$	$T = 100$	$T = 400$
SIR		0.9849 (0.0011) 0.94		
$OSIR_2$		0.9855 (0.0012) 0.97		
$OSIR_5$		0.9847 (0.0012) 0.99		
$OSIR_7$		0.9853 (0.0012) 1.00		
$OSIR_9$		0.9854 (0.0011) 1.00		
CUME		0.9838 (0.0013) 1.00		
Bagging-I	SIR	0.9775 (0.0015) 0.79	0.9860 (0.0011) 0.84	0.9864 (0.0010) 0.86
	$OSIR_2$	0.9809 (0.0015) 0.91	0.9859 (0.0011) 0.95	0.9862 (0.0011) 0.95
	$OSIR_5$	0.9786 (0.0016) 0.97	0.9845 (0.0013) 0.99	0.9855 (0.0012) 0.99
	$OSIR_7$	0.9801 (0.0013) 0.99	0.9852 (0.0012) 0.99	0.9863 (0.0011) 0.99
	$OSIR_9$	0.9784 (0.0019) 0.98	0.9858 (0.0012) 0.99	0.9862 (0.0011) 0.99
	CUME	0.9785 (0.0016) 1.00	0.9836 (0.00124) 1.00	0.9842 (0.0012) 1.00
Bagging-II	SIR	0.9851 (0.0010) 0.05	0.9866 (0.0012) 0.02	0.9866 (0.0010) 0.02
	$OSIR_2$	0.9856 (0.0011) 0.85	0.9860 (0.0010) 0.87	0.9863 (0.0011) 0.88
	$OSIR_5$	0.9843 (0.0012) 0.98	0.9855 (0.0011) 0.99	0.9857 (0.0011) 0.98
	$OSIR_7$	0.9851 (0.0011) 0.99	0.9860 (0.0011) 0.99	0.9862 (0.0011) 0.99
	$OSIR_9$	0.9840 (0.0012) 0.98	0.9861 (0.0011) 0.99	0.9862 (0.0011) 0.99
	CUME	0.9822 (0.0013) 1.00	0.9844 (0.0012) 1.00	0.9840 (0.0012) 1.00
Bootstrapping	SIR	0.9822 (0.0014) 0.80	0.9855 (0.0011) 0.87	0.9850 (0.0011) 0.87
	$OSIR_2$	0.9825 (0.0016) 0.94	0.9844 (0.0013) 0.95	0.9851 (0.0012) 0.95
	$OSIR_5$	0.9818 (0.0014) 1.00	0.9842 (0.0013) 0.98	0.9844 (0.0013) 0.98
	$OSIR_7$	0.9819 (0.00126) 0.99	0.9844 (0.0013) 0.99	0.9847 (0.0013) 0.99
	$OSIR_9$	0.9832 (0.0015) 0.99	0.9844 (0.0013) 0.98	0.9851 (0.0012) 0.98
	CUME	0.9803 (0.0014) 1.00	0.9825 (0.0014) 1.00	0.9824 (0.0016) 1.00

100 times. To see the impact of bootstrapping, we run the simulations by varying T values from $\{10, 100, 400\}$.

Tables 3.1, 3.2, 3.3 and 3.4 show the results of the trace correlation mean, trace correlation standard error, and the predication accuracy of the intrinsic dimensionality. We see that both bootstrapping techniques help improve the performance of OSIR while these two techniques does not show significant difference. Again, we see that the overlapping level $L = H/2$ allows to get the best EDR space and dimensionality prediction.

Table 3.2. Accuracy of EDR space estimation for Model (2.3).

Algorithm \ Replicates		$T = 10$	$T = 100$	$T = 400$
SIR		0.7357 (0.0187) 0.08		
$OSIR_2$		0.8039 (0.0128) 0.33		
$OSIR_5$		0.8148 (0.0121) 0.54		
$OSIR_7$		0.8179 (0.0120) 0.60		
$OSIR_9$		0.8182 (0.0120) 0.56		
CUME		0.8179 (0.0120) 1.00		
Bagging-I	SIR	0.7221 (0.0201) 0.00	0.7654 (0.0162) 0.00	0.7734 (0.0163) 0.00
	$OSIR_2$	0.8024 (0.0136) 0.02	0.8075 (0.0125) 0.02	0.8137 (0.0121) 0.02
	$OSIR_5$	0.7991 (0.0133) 0.08	0.8174 (0.0121) 0.10	0.8192 (0.0119) 0.11
	$OSIR_7$	0.8026 (0.0127) 0.07	0.8250 (0.0117) 0.07	0.8228 (0.0116) 0.10
	$OSIR_9$	0.8020 (0.0132) 0.04	0.8253 (0.0118) 0.06	0.8228 (0.0116) 0.06
	CUME	0.8017 (0.0129) 1.00	0.8140 (0.0121) 1.00	0.8164 (0.0121) 1.00
Bagging-II	SIR	0.7143 (0.0200) 0.00	0.7683 (0.0171) 0.00	0.7707 (0.0155) 0.00
	$OSIR_2$	0.7998 (0.0134) 0.00	0.8131 (0.0122) 0.00	0.8149 (0.0121) 0.00
	$OSIR_5$	0.8077 (0.0132) 0.07	0.8203 (0.0116) 0.09	0.8197 (0.0121) 0.07
	$OSIR_7$	0.8153 (0.0121) 0.15	0.8215 (0.0118) 0.10	0.8236 (0.0116) 0.09
	$OSIR_9$	0.8067 (0.0118) 0.09	0.8222 (0.0118) 0.05	0.8232 (0.0117) 0.04
	CUME	0.8046 (0.0125) 1.00	0.8171 (0.0119) 1.00	0.8184 (0.0121) 1.00
Bootstrapping	SIR	0.7119 (0.0201) 0.00	0.7663 (0.0175) 0.00	0.7703 (0.0169) 0.00
	$OSIR_2$	0.7853 (0.0153) 0.01	0.8141 (0.0131) 0.04	0.8168 (0.0122) 0.04
	$OSIR_5$	0.7872 (0.0147) 0.06	0.8215 (0.0120) 0.10	0.8217 (0.0123) 0.10
	$OSIR_7$	0.8062 (0.0131) 0.08	0.8227 (0.0124) 0.11	0.8247 (0.0122) 0.09
	$OSIR_9$	0.7866 (0.0170) 0.02	0.8224 (0.0125) 0.03	0.8246 (0.0120) 0.05
	CUME	0.7876 (0.0149) 1.00	0.8200 (0.0118) 1.00	0.8167 (0.0123) 1.00

3.4.2 Real data application

We again use the Boston housing price data to test the ability of bagging OSIR methods to extract the EDR space. We follow the same simulation strategy as in Chapter 2 but set the number of observations in training set to be 100, skip the correlation calculation and only report the mean and standard error of the prediction accuracy for each method. We choose the replicates T from $\{10, 100, 400\}$. The simulation results are reported in Table 3.5 and Figure 3.2, which confirmed the advantages of using bagging and the alternative bootstrapping techniques to improve

Table 3.3. Accuracy of EDR space estimation for Model (2.4).

Algorithm \ Replicates		$T = 10$	$T = 100$	$T = 400$
SIR		0.7253 (0.0133) 0.40		
$OSIR_2$		0.7909 (0.0095) 0.88		
$OSIR_5$		0.8021 (0.0088) 0.98		
$OSIR_7$		0.8058 (0.0087) 0.98		
$OSIR_9$		0.8042 (0.0088) 0.96		
CUME		0.7895 (0.0092) 0.00		
Bagging-I	SIR	0.7206 (0.0135) 0.17	0.7520 (0.0122) 0.23	0.7517 (0.0123) 0.22
	$OSIR_2$	0.7788 (0.0098) 0.53	0.7944 (0.0094) 0.66	0.7964 (0.0093) 0.66
	$OSIR_5$	0.7835 (0.0096) 0.76	0.8050 (0.0087) 0.90	0.8062 (0.0086) 0.91
	$OSIR_7$	0.7874 (0.0090) 0.76	0.8077 (0.0086) 0.87	0.8099 (0.0085) 0.89
	$OSIR_9$	0.7833 (0.0099) 0.58	0.8104 (0.0088) 0.80	0.8104 (0.0086) 0.81
	CUME	0.7671 (0.0099) 0.00	0.7936 (0.0091) 0.00	0.7940 (0.0090) 0.00
Bagging-II	SIR	0.6953 (0.0141) 0.00	0.7413 (0.0122) 0.00	0.7426 (0.0125) 0.00
	$OSIR_2$	0.7713 (0.0104) 0.42	0.7872 (0.0099) 0.39	0.7903 (0.0095) 0.39
	$OSIR_5$	0.7843 (0.0092) 0.95	0.7969 (0.0090) 0.94	0.7997 (0.0089) 0.97
	$OSIR_7$	0.7915 (0.0094) 0.95	0.8033 (0.0085) 0.98	0.8029 (0.0088) 0.98
	$OSIR_9$	0.7875 (0.0095) 0.91	0.7981 (0.0091) 0.90	0.8026 (0.0089) 0.91
	CUME	0.7725 (0.0095) 0.00	0.7860 (0.0093) 0.00	0.7868 (0.0092) 0.00
Bootstrapping	SIR	0.7173 (0.0126) 0.33	0.7573 (0.0123) 0.28	0.7575 (0.0125) 0.27
	$OSIR_2$	0.7683 (0.0116) 0.67	0.7931 (0.0100) 0.72	0.7970 (0.0095) 0.69
	$OSIR_5$	0.7640 (0.0109) 0.78	0.8051 (0.0088) 0.94	0.8099 (0.0087) 0.90
	$OSIR_7$	0.7680 (0.0110) 0.78	0.8127 (0.0088) 0.91	0.8124 (0.0086) 0.89
	$OSIR_9$	0.7635 (0.0109) 0.72	0.8106 (0.0085) 0.83	0.8121 (0.0085) 0.84
	CUME	0.7504 (0.0113) 0.00	0.7907 (0.0095) 0.00	0.7981 (0.0088) 0.00

the EDR estimations.

3.5 Conclusions and discussions

We have proved the effectiveness of the overlapping strategy in the last chapter. In this chapter we explored the application of bootstrapping with simple output averaging or extended Jacobian angles for simultaneous diagonalization in the SIR-like algorithms. Both overlapping and bootstrapping are shown to be able to retrieve a better estimate of the EDR space with limited sample size. To close this chapter,

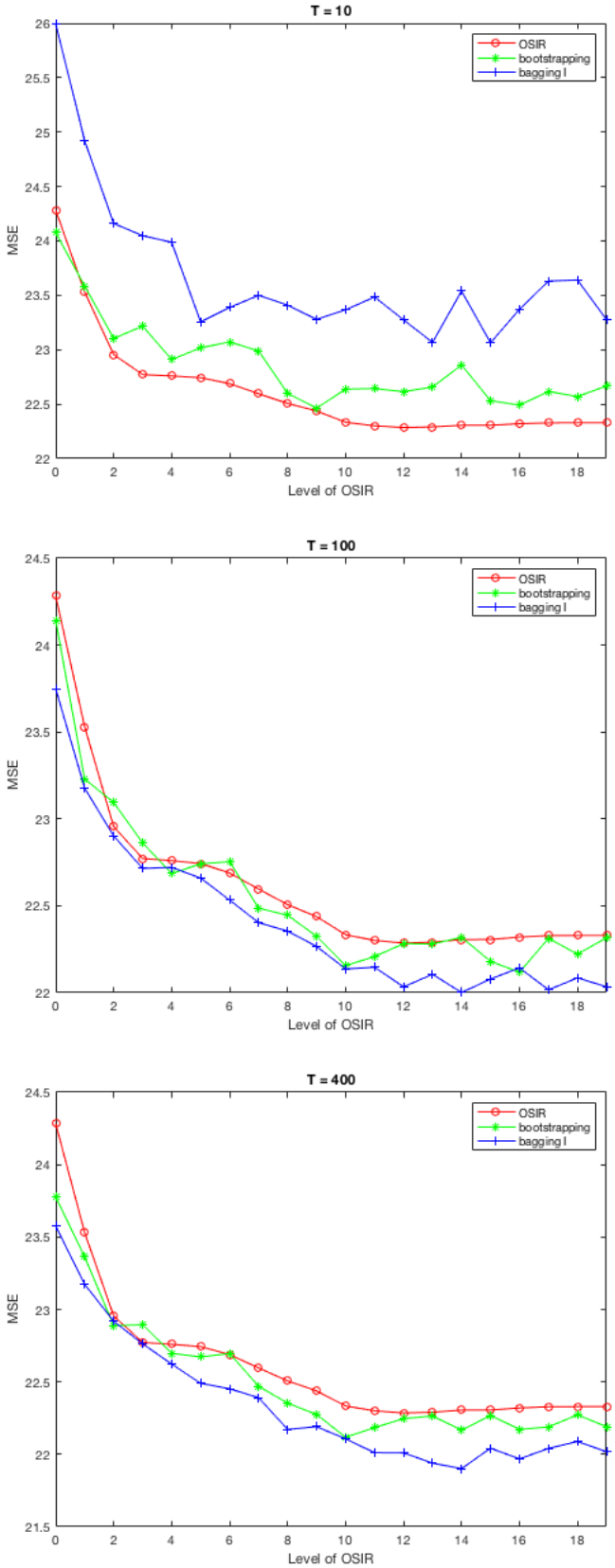


Figure 3.2. Comparison among OSIR, bagging-I OSIR and bootstrapping OSIR.

Table 3.4. Accuracy of EDR space estimation for Model (2.5).

Algorithm \ Replicates		$T = 10$	$T = 100$	$T = 400$
SIR		0.7320 (0.0118) 0.48		
$OSIR_2$		0.7884 (0.0086) 0.90		
$OSIR_5$		0.8050 (0.0076) 0.97		
$OSIR_7$		0.8086 (0.0073) 0.96		
$OSIR_9$		0.8061 (0.0075) 0.95		
CUME		0.7870 (0.0087) 0.00		
Bagging-I	SIR	0.7319 (0.0112) 0.10	0.7673 (0.0102) 0.21	0.7693 (0.0100) 0.20
	$OSIR_2$	0.7720 (0.0095) 0.62	0.7929 (0.0085) 0.64	0.7934 (0.0085) 0.66
	$OSIR_5$	0.7908 (0.0083) 0.93	0.8075 (0.0074) 0.93	0.8112 (0.0074) 0.95
	$OSIR_7$	0.7959 (0.0082) 0.87	0.8114 (0.0071) 0.95	0.8142 (0.0072) 0.96
	$OSIR_9$	0.7926 (0.0076) 0.74	0.8088 (0.0077) 0.92	0.8112 (0.0074) 0.92
	CUME	0.7618 (0.0096) 0.00	0.7837 (0.0087) 0.00	0.7901 (0.0084) 0.00
Bagging-II	SIR	0.7292 (0.0118) 0.01	0.7642 (0.0101) 0.00	0.7618 (0.0108) 0.00
	$OSIR_2$	0.7676 (0.0088) 0.47	0.7890 (0.0089) 0.40	0.7902 (0.0088) 0.42
	$OSIR_5$	0.7894 (0.0077) 0.99	0.8046 (0.0077) 0.99	0.8067 (0.0076) 0.98
	$OSIR_7$	0.7977 (0.0086) 0.99	0.8106 (0.0074) 0.98	0.8116 (0.0073) 0.98
	$OSIR_9$	0.7932 (0.0088) 0.95	0.8061 (0.0076) 0.96	0.8089 (0.0075) 0.97
	CUME	0.7563 (0.0094) 0.00	0.7822 (0.0090) 0.00	0.7838 (0.0088) 0.00
Bootstrapping	SIR	0.7173 (0.0126) 0.33	0.7573 (0.0123) 0.28	0.7575 (0.0125) 0.27
	$OSIR_2$	0.7683 (0.0116) 0.67	0.7931 (0.0100) 0.72	0.7970 (0.0095) 0.69
	$OSIR_5$	0.7640 (0.0109) 0.78	0.8051 (0.0088) 0.94	0.8099 (0.0087) 0.90
	$OSIR_7$	0.7680 (0.0110) 0.78	0.8127 (0.0088) 0.91	0.8124 (0.0086) 0.89
	$OSIR_9$	0.7635 (0.0109) 0.72	0.8106 (0.0085) 0.83	0.8121 (0.0085) 0.84
	CUME	0.7504 (0.0113) 0.00	0.7907 (0.0095) 0.00	0.7981 (0.0088) 0.00

we remark that the benefits brought by bootstrapping diminishes as the sample size increase. When the sample size is very large and the EDR estimation by SIR or OSIR is already very accurate, bootstrapping does not help much. So bootstrapping is recommended when the sample size is relatively small and the EDR estimation from SIR or OSIR is not as good. Also, it seems there is no promise that one bootstrapping strategy outperforms another all the time.

Table 3.5. Prediction accuracy on Boston housing price data.

Algorithm \ Replicates		$T = 10$	$T = 100$	$T = 400$
SIR			24.2831 (0.4009)	
$OSIR_2$			22.9554 (0.3355)	
$OSIR_3$			22.7722 (0.3278)	
$OSIR_5$			22.7433 (0.3266)	
$OSIR_{10}$			22.3336 (0.3121)	
$OSIR_{15}$			22.3059 (0.3127)	
$OSIR_{19}$			22.3294 (0.3037)	
CUME			23.5774 (0.3064)	
KNN			53.2681 (0.5463)	
MLR			23.2193 (0.3248)	
Bagging-I	SIR	25.9968 (0.4280)	23.7489 (0.3473)	23.5766 (0.3527)
	$OSIR_2$	24.1607 (0.3720)	22.9018 (0.3215)	22.9187 (0.3384)
	$OSIR_3$	24.0479 (0.3950)	22.7149 (0.3266)	22.7626 (0.3424)
	$OSIR_5$	23.2559 (0.3620)	22.6596 (0.3307)	22.4901 (0.3068)
	$OSIR_{10}$	23.3666 (0.3536)	22.1370 (0.3088)	22.1066 (0.3072)
	$OSIR_{15}$	23.0640 (0.3288)	22.0789 (0.3146)	22.0422 (0.2988)
	$OSIR_{19}$	23.2770 (0.3709)	22.0343 (0.3097)	22.0184 (0.2965)
	CUME	22.9611 (0.3459)	23.1063 (0.3471)	23.1257 (0.3505)
Bagging-II	SIR	36.1423 (1.6052)	44.2176 (2.7331)	55.4316 (3.2356)
	$OSIR_2$	33.6886 (1.5477)	44.5956 (2.8699)	53.3035 (3.1676)
	$OSIR_3$	34.7295 (1.5705)	40.5122 (2.4379)	50.9358 (3.1648)
	$OSIR_5$	35.2912 (1.5855)	41.5326 (2.5864)	56.0961 (3.3180)
	$OSIR_{10}$	33.5910 (1.4478)	41.2416 (2.4549)	54.1709 (3.0798)
	$OSIR_{15}$	33.0274 (1.4159)	39.5041 (2.6061)	48.8280 (3.1242)
	$OSIR_{19}$	32.6730 (1.4130)	37.5451 (2.4310)	53.7906 (3.3623)
	CUME	54.9420 (3.2753)	52.6485 (3.2114)	54.5268 (3.3410)
Bootstrapping	SIR	24.0767 (0.4126)	24.1421 (0.3943)	23.7772 (0.3418)
	$OSIR_2$	23.1040 (0.3586)	23.0941 (0.3414)	22.8884 (0.3278)
	$OSIR_3$	23.2166 (0.3357)	22.8615 (0.3429)	22.8949 (0.3270)
	$OSIR_5$	23.0175 (0.2692)	22.7414 (0.3189)	22.6726 (0.3051)
	$OSIR_{10}$	22.6379 (0.3207)	22.1560 (0.2986)	22.1181 (0.2923)
	$OSIR_{15}$	22.5323 (0.3233)	22.1805 (0.3166)	22.2687 (0.3099)
	$OSIR_{19}$	22.6673 (0.3322)	22.3158 (0.3077)	22.1908 (0.3164)
	CUME	23.1698 (0.3418)	23.2259 (0.3617)	23.1281 (0.3572)

CHAPTER 4

INCREMENTAL SLICED INVERSE REGRESSION

4.1 Introduction

Due to the fast development of modern information technology, we are in a big data era and facing the challenges of big data processing, among which two primary challenges are the big volume and fast velocity of the data. When a data set is too big to store in a single machine or when the data arrives in real time and information update is needed frequently, analysis of the data in an online manner is necessary and efficient. If the data is simultaneously big and high dimensional, it becomes necessary to develop incremental learning approaches for dimension reduction. To be more specific, we define an incremental learning for dimension reduction as one that meets the following criteria according to a definition of the incremental learning algorithm for classification [89]:

1. It should be able to learn additional information from new data.
2. It should not require to access to the original data to train the existing model.
3. It should preserve the previously acquired knowledge.
4. It should be able to accommodate new dimensionality that may be introduced with new data.

As PCA and LDA are most wildly used in dimension reduction techniques, correspondingly, a bunch of PCA-based and LDA-based online dimension reduction algorithms have been proposed. Incremental PCA have been described in [17, 45, 46,

90, 91, 102, 122]. Incremental LDA have been developed in [22, 42, 63, 67, 88, 97, 99, 108, 110, 112, 121]. Although many SIR related algorithms have been proposed, to our best knowledge, there is not incremental learning method for incremental supervised dimension reduction in the regression setting that satisfies the criteria above. In this chapter, our purpose is to propose such a new incremental learning approach. The new data can show up by a single observation or by block. We are going to figure out how to update an SIR model when a single observation is received. For block update, we can take it as several single observation problems temporarily.

Our motivation is to implement the sliced inverse regression (SIR) in an incremental manner. SIR can be implemented by solving an generalized eigen-decomposition problem in Equation (1.5). To make it implementable in an online manner we rewrite it as normal eigen-decomposition problem $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\eta} = \lambda\boldsymbol{\eta}$, where $\boldsymbol{\eta} = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{\beta}$ and adopt the ideas from incremental PCA. We need to overcome two main challenges in this process. First, how do we transform the data so that they are appropriate for the transformed PCA problem? Note that simply normalizing the data does not work. Second, online update of $\boldsymbol{\Sigma}^{-\frac{1}{2}}$, if not impossible, seems very difficult, The first contribution of this chapter is to overcome these difficulties and design a workable incremental SIR method. Our second contribution will be to refine the method by an overlapping technique and design an incremental overlapping SIR algorithm.

The rest of this chapter is arranged as follows. We review the incremental PCA algorithm in Section 4.2. We propose the incremental SIR algorithm in Section 4.3 and refine it in Section 4.4. Simulations are done in Section 4.5 and we close with discussions in Section 4.6 and provide the formula inference in Section 4.7.

4.2 Incremental PCA

PCA looks for directions along which the data have largest variances. It is implemented by solving an eigen-decomposition problem

$$\widehat{\mathbf{\Sigma}}\mathbf{u} = \lambda\mathbf{u}. \tag{4.1}$$

The principal components are the eigenvectors corresponding to largest eigenvalues. Throughout this chapter, we assume all eigenvalues are arranged in a descending order, i.e., $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$. Suppose that we need to retain the top K principal components, denote $\mathbf{U}_K = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$, $\mathbf{\Lambda}_K = \text{diag}(\lambda_1, \ldots, \lambda_K)$, we have a reduced system $\widehat{\mathbf{\Sigma}}\mathbf{U}_K = \mathbf{U}_K\mathbf{\Lambda}_K$.

In incremental PCA, after receiving a new coming observation $\mathbf{x}_{n+1}$, we need to update the reduced eigen-system to a new one

$$\widehat{\mathbf{\Sigma}}'\mathbf{U}'_K \approx \mathbf{U}'_K\mathbf{\Lambda}'_K. \tag{4.2}$$

The idea of updating the system in [45] is as follows. Compute a residual vector

$$\mathbf{v} = (\mathbf{x}_{n+1} - \bar{\mathbf{x}}') - \mathbf{U}_K\mathbf{U}_K^\top(\mathbf{x}_{n+1} - \bar{\mathbf{x}}'),$$

where $\bar{\mathbf{x}}'$ is the mean of all observations (including $\mathbf{x}_{n+1}$). It defines the component of $\mathbf{x}_{n+1}$ that is perpendicular with the subspace defined by $\mathbf{U}_K$. If $\mathbf{x}_{n+1}$ lies exactly within the current eigenspace, then the residual vector is zero and there is no need to update the system. Otherwise, we normalize $\mathbf{v}$ to obtain $\bar{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$. We may reasonably assume each column vector of $\mathbf{U}'_K$ is a linear combination of column vectors of $\mathbf{U}_K$ and $\bar{\mathbf{v}}$ – this is exactly true if $\lambda_{K+1} = \ldots = \lambda_p = 0$. This allows us to write

$$[\mathbf{U}'_K, \mathbf{u}'_{K+1}] = [\mathbf{U}_K, \bar{\mathbf{v}}]\mathbf{R}.$$

where $\mathbf{R}$ is a $(K+1)\times(K+1)$ rotation matrix and $\mathbf{u}'_{K+1}$ is an approximation of the $(K+1)$th eigenvector of $\widehat{\mathbf{\Sigma}}'$. So we have

$$\widehat{\mathbf{\Sigma}}'[\mathbf{U}_K, \bar{\mathbf{v}}]\mathbf{R} = [\mathbf{U}_K, \bar{\mathbf{v}}]\mathbf{R}\mathbf{\Lambda}'_{K+1},$$

which is equivalent to

$$[\mathbf{U}_K, \bar{\mathbf{v}}]^\top \widehat{\mathbf{\Sigma}}' [\mathbf{U}_K, \widehat{\mathbf{v}}]\mathbf{R} = \mathbf{R}\mathbf{\Lambda}'_{K+1}.$$

This is an eigen-decomposition problem of dimensionality $K+1 \ll p$. It solves the rotation matrix $\mathbf{R}$ and allows us to update principal components to $\mathbf{U}'_K$, given by the first K columns of $[\mathbf{U}_k, \bar{\mathbf{v}}]\mathbf{R}$. If we need to increase the number of principal components, we can just update the system to $K' = K+1$ and $\mathbf{U}'_{K'} = [\mathbf{U}'_K, \mathbf{u}'_{K+1}]$. This incremental PCA algorithm was shown convergent to a stable solution when the sample size increases [45].

4.3 Incremental SIR

Our idea to develop the incremental sliced inverse regression (ISIR) is motivated by reformulating SIR problem to a PCA problem. To this end, we define $\boldsymbol{\eta} = \mathbf{\Sigma}^{\frac{1}{2}}\boldsymbol{\beta}$, called the standardized EDR direction, and rewrite the generalized eigen-decomposition problem (1.5) as an eigen-decomposition problem

$$\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{\Gamma}\mathbf{\Sigma}^{-\frac{1}{2}}\boldsymbol{\eta} = \lambda\boldsymbol{\eta}. \tag{4.3}$$

Note that $\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{\Gamma}\mathbf{\Sigma}^{-\frac{1}{2}}$ is the covariance matrix of $\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{E}[\mathbf{x}|y]$. So Equation (4.3) can be regarded as a PCA problem with data collected for $\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{E}[\mathbf{x}|y]$. To apply the ideas from IPCA to this transformed PCA problem, however, is not as direct as it looks like. We face two main challenges. First, when a new observation $(\mathbf{x}_{n+1}, y_{n+1})$ is received, we need to transform it to an observation for the standardized inverse regression

curve. This is different from simply standardizing the data. Second, conceptually, we need to update $\mathbf{\Sigma}^{-\frac{1}{2}}$ in an online manner in order to standardize the data. This does not seem feasible. In the following, we will describe in detail how we address these challenges and make the ISIR implementable.

Suppose we have n observations in hand with well defined sample slice probabilities $\hat{p}_h$ and means $(\hat{\mathbf{m}}_h, \bar{y}_h)$ for $h = 1, \ldots, H$, and the eigenvectors $\widehat{\mathbf{B}} = [\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_K]$ of the generalized eigen-decomposition problem $\widehat{\mathbf{\Gamma}}\boldsymbol{\beta} = \lambda\widehat{\mathbf{\Sigma}}\boldsymbol{\beta}$. With $\mathbf{\Lambda}_K = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$, we have

$$\widehat{\mathbf{\Gamma}}\widehat{\mathbf{B}} = \widehat{\mathbf{\Sigma}}\widehat{\mathbf{B}}\mathbf{\Lambda}_K.$$

Denote $\mathbf{\Xi} = \widehat{\mathbf{\Sigma}}^{\frac{1}{2}}\widehat{\mathbf{B}}$. We have

$$\widehat{\mathbf{\Sigma}}^{-\frac{1}{2}}\widehat{\mathbf{\Gamma}}\widehat{\mathbf{\Sigma}}^{-\frac{1}{2}}\mathbf{\Xi} = \mathbf{\Xi}\mathbf{\Lambda}_K. \tag{4.4}$$

When we have a new observation $(\mathbf{x}_{n+1}, y_{n+1})$, we first locate which slice it belongs to according to the distances from y_{n+1} to sample slice mean values $\bar{y}_h$ of the response variable. Let us suppose the distance from y_{n+1} to $\bar{y}_s$ is the smallest. So we place the new observation into the slice s and update sample slice probabilities by $\hat{p}'_h = \frac{n\hat{p}_h}{n+1}$ for $h \neq s$ and $p'_s = \frac{np_s+1}{n+1}$. Let $n_s = np_s$ be the number of observations in slice s before receiving the new observation. For slice mean values we update

$$\mathbf{m}'_s = \frac{n_s}{n_s+1}\mathbf{m}_s + \frac{1}{n_s+1}\mathbf{x}_{n+1}$$

for slice s only. We can regard $\mathbf{z}_{n+1} = \widehat{\mathbf{\Sigma}}^{-\frac{1}{2}}\mathbf{m}'_s$ as a new observation for the standardized inverse regression curve $\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{E}[\mathbf{x}|y]$. Following the idea of IPCA, we define a residual vector

$$\mathbf{v} = \left(\mathbf{z}_{n+1} - \widehat{\mathbf{\Sigma}}^{-\frac{1}{2}}\bar{\mathbf{x}}'\right) - \mathbf{\Xi}\mathbf{\Xi}^\top\left(\mathbf{z}_{n+1} - \widehat{\mathbf{\Sigma}}^{-\frac{1}{2}}\bar{\mathbf{x}}'\right)$$

and normalize it to $\bar{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ when $\mathbf{v}$ is not zero. To update the eigen-decomposition system to

$$\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}} \widehat{\boldsymbol{\Gamma}}' \widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}} \boldsymbol{\Xi}' = \boldsymbol{\Xi}' \boldsymbol{\Lambda}'_K, \tag{4.5}$$

we assume $[\boldsymbol{\Xi}', \boldsymbol{\eta}'_{K+1}] = [\boldsymbol{\Xi}, \bar{\mathbf{v}}]\mathbf{R}$ with $\mathbf{R}$ being a $(K+1) \times (K+1)$ rotation matrix and $\boldsymbol{\eta}_{K+1}$ the $(K+1)$th eigenvector of $\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}} \widehat{\boldsymbol{\Gamma}}' \widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}$. So we have

$$\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}} \widehat{\boldsymbol{\Gamma}}' \widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}} [\boldsymbol{\Xi}, \bar{\mathbf{v}}]\mathbf{R} = [\boldsymbol{\Xi}, \bar{\mathbf{v}}]\mathbf{R}\boldsymbol{\Lambda}'_{K+1}$$

where $\boldsymbol{\Lambda}'_{K+1} = \mathrm{diag}(\boldsymbol{\Lambda}'_K, \lambda'_{K+1})$ and λ'_{K+1} is the $(K+1)$th eigenvalue. Multiplying both sides by $[\boldsymbol{\Xi}, \bar{\mathbf{v}}]^\top$, we obtain

$$\left(\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}} [\boldsymbol{\Xi}, \bar{\mathbf{v}}]\right)^\top \widehat{\boldsymbol{\Gamma}}' \left(\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}} [\boldsymbol{\Xi}, \bar{\mathbf{v}}]\right) \mathbf{R} = \mathbf{R}\boldsymbol{\Lambda}'_{K+1}. \tag{4.6}$$

Note that $\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}$ cannot be easily updated, we have to avoid using it. To overcome this challenge, we notice that

$$\widehat{\boldsymbol{\Sigma}'} = \frac{n}{n+1}\widehat{\boldsymbol{\Sigma}} + \frac{n}{(n+1)^2}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})(\mathbf{x}_{n+1} - \bar{\mathbf{x}})^\top \tag{4.7}$$

and the well known Sherman-Morrison formula allows us to update the inverse matrix

$$\widehat{\boldsymbol{\Sigma}'}^{-1} = \frac{n+1}{n}\widehat{\boldsymbol{\Sigma}}^{-1} - \frac{\frac{1}{n}\widehat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})(\mathbf{x}_{n+1} - \bar{\mathbf{x}})^\top \widehat{\boldsymbol{\Sigma}}^{-1}}{1 + \frac{1}{n+1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})^\top \widehat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})}. \tag{4.8}$$

See Section 4.7 for the detailed proof of (4.8). We just need to compute and store $\boldsymbol{\Sigma}'^{-1}$ once and there is no more need to do inverse matrix calculation which is always time consuming and inaccurate numerically. If we store $\widehat{\boldsymbol{\Sigma}}^{-1}$ and update it incrementally,

we can approximate the quantities in (4.6) as follows:

$$\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}\boldsymbol{\Xi} \approx \widehat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}\boldsymbol{\Xi} = \widehat{\mathbf{B}},$$

$$\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}\mathbf{v} \approx \widehat{\boldsymbol{\Sigma}}^{-1}\left(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}'\right) - \widehat{\mathbf{B}}\widehat{\mathbf{B}}^{\top}\left(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}'\right),$$

$$\|\mathbf{v}\|^2 = \left(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}'\right)^{\top}\left(\widehat{\boldsymbol{\Sigma}}^{-1} - \widehat{\mathbf{B}}\widehat{\mathbf{B}}^{\top}\right)\left(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}'\right),$$

$$\begin{aligned}\tilde{\mathbf{v}} = \widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}\bar{\mathbf{v}} &= \frac{\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}\mathbf{v}}{\|\mathbf{v}\|} \\ &\approx \frac{\left(\widehat{\boldsymbol{\Sigma}}^{-1} - \widehat{\mathbf{B}}\widehat{\mathbf{B}}^{\top}\right)\left(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}'\right)}{\sqrt{\left(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}'\right)^{\top}\left(\widehat{\Sigma}^{-1} - \widehat{\mathbf{B}}\widehat{\mathbf{B}}^{\top}\right)\left(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}'\right)}}.\end{aligned}$$

So the problem (4.6) is approximated by

$$\left[\widehat{\mathbf{B}}, \tilde{\mathbf{v}}\right]^{\top}\widehat{\boldsymbol{\Gamma}'}\left[\widehat{\mathbf{B}}, \tilde{\mathbf{v}}\right]\mathbf{R} = \mathbf{R}\boldsymbol{\Lambda}'_{K+1}. \tag{4.9}$$

Finally notice that the new EDR space $\widehat{\mathbf{B}}' = \widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}\boldsymbol{\Xi}'$ is the first K columns of $\widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}[\boldsymbol{\Xi}', \boldsymbol{\eta}'_{K+1}] = \widehat{\boldsymbol{\Sigma}'}^{-\frac{1}{2}}[\boldsymbol{\Xi}, \bar{\mathbf{v}}]\mathbf{R}$ and can be approximated by the first K columns of $[\widehat{\mathbf{B}}, \tilde{\mathbf{v}}]\mathbf{R}$.

Note that we avoided updating the inverse square root of the covariance matrix by using the approximation $\widehat{\boldsymbol{\Sigma}}^{-\frac{1}{2}} \approx \widehat{\boldsymbol{\Sigma}}'^{-\frac{1}{2}}$. This approximation can be very accurate when n is large enough because both converge to $\boldsymbol{\Sigma}^{-\frac{1}{2}}$. Therefore, we may expect the convergence of ISIR as a corollary of the convergence of IPCA. However, when n is small, the approximation may be less accurate and result in larger difference between EDR spaces estimated by ISIR and SIR. So we recommend that ISIR be used with a warm start, that is, using SIR first on a small amount of data before using ISIR.

In terms of memory, the primary requirement is the storage of $\widehat{\boldsymbol{\Sigma}}^{-1}$, the slice mean

matrix $\widehat{\mathbf{M}} = [\hat{\mathbf{m}}_1, \ldots, \hat{\mathbf{m}}_H]$, and the EDR space $\widehat{\mathbf{B}}$. So the memory requirement is $O(p^2+pH+pK)$. As for the computational complexity, notice that the update of $\widehat{\mathbf{\Sigma}'}^{-1}$ in Equation (4.8) requires the calculation of $\widehat{\mathbf{\Sigma}}^{-1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})$ and matrix addition and has a complexity of $O(p^2)$. Since we need to store $\widehat{\mathbf{M}}$ and update it sequentially, it is not efficient to store and update $\widehat{\mathbf{\Gamma}}'$ for either memory or computation consideration. Instead, we use the fact $\widehat{\mathbf{\Gamma}}' = \widehat{\mathbf{M}}'\widehat{\mathbf{P}}'\widehat{\mathbf{M}}'^{\top}$ where $\widehat{\mathbf{P}}' = \mathrm{diag}(\hat{p}'_1, \ldots, \hat{p}'_H)$ and write

$$[\widehat{\mathbf{B}}, \tilde{\mathbf{v}}]^{\top}\widehat{\mathbf{\Gamma}}'[\widehat{\mathbf{B}}, \tilde{\mathbf{v}}] = \left[\widehat{\mathbf{B}}^{\top}\widehat{\mathbf{M}}', \tilde{\mathbf{v}}^{\top}\widehat{\mathbf{M}}'\right] \widehat{\mathbf{P}}' \left[\widehat{\mathbf{B}}^{\top}\widehat{\mathbf{M}}', \tilde{\mathbf{v}}^{\top}\widehat{\mathbf{M}}'\right]^{\top}.$$

Notice that

$$\tilde{\mathbf{v}}^{\top}\widehat{\mathbf{M}}' = \frac{\left(\widehat{\mathbf{\Sigma}}^{-1}(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')\right)^{\top}\widehat{\mathbf{M}}' - \left(\widehat{\mathbf{B}}^{\top}(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')\right)^{\top}\left(\widehat{\mathbf{B}}^{\top}\widehat{\mathbf{M}}'\right)}{\sqrt{(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')^{\top}\left(\widehat{\mathbf{\Sigma}}^{-1}(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')\right) - \left(\widehat{\mathbf{B}}^{\top}(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')\right)^{\top}\left(\widehat{\mathbf{B}}^{\top}(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')\right)}}$$

and $\widehat{\mathbf{B}}^{\top}(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')$ is just the sth column of $\widehat{\mathbf{B}}^{\top}\widehat{\mathbf{M}}'$. The primary computation for the matrix $[\widehat{\mathbf{B}}, \tilde{\mathbf{v}}]^{\top}\widehat{\mathbf{\Gamma}}'[\widehat{\mathbf{B}}, \tilde{\mathbf{v}}]$ is $\widehat{\mathbf{B}}^{\top}\widehat{\mathbf{M}}'$ and $\widehat{\mathbf{\Sigma}}^{-1}(\hat{\mathbf{m}}'_s - \bar{\mathbf{x}}')$ which has a complexity of $O(pKH + p^2)$. The complexity of the eigen-decomposition in Equation (4.9) is $O((K+1)^3)$ and to update $\widehat{\mathbf{B}}$ to $\widehat{\mathbf{B}}'$ requires $O(p(K+1)^2)$. So the computational complexity for the whole ISIR update is $O(p^2 + pKH + pK^2 + K^3)$. For a high dimensional problem, this is much smaller than the complexity of $O(p^3 + p^2 n)$ for SIR.

4.4 Refinement by overlapping

In Chapter 2, an overlapping technique was introduced to SIR algorithm and shown effectively improving the accuracy of EDR space estimation. It is motivated by placing each observation in two or more adjacent slices to reduce the deviations of the sample slice means $\hat{\mathbf{m}}_h$ from the EDR subspace. This is equivalent to using each observation two or more times. In this section, we adopt the overlapping technique to

ISIR algorithm above to develop an incremental overlapping sliced inverse regression (IOSIR) algorithm and wish it refines ISIR.

To apply the overlapping idea, we use each observation twice. So when we have n observations, we duplicate them and assume we have $N = 2n$ observations. When a new observation $(\mathbf{x}_{n+1}, y_{n+1})$ is received, we duplicate it and assume we receive two identical observations. Based on the y_{n+1} value we place the first copy into the slice s if $\bar{y}_s$ is the closest to y_{n+1} and run ISIR update as described in Section 4.3. Note that if $\bar{y}_1 < y_{n+1} < \bar{y}_H$, then y_{n+1} must fall into the interval $[\bar{y}_{s'}, \bar{y}_s]$ with $s' = s - 1$ or it falls into $[\bar{y}_s, \bar{y}_{s'}]$ with $s' = s + 1$. So we place the second copy of the new observation to slice s', which is adjacent to slice s, and run ISIR algorithm again. If $y_{n+1} \leq \bar{y}_1$ or $y_{n+1} > \bar{y}_H$, the second copy will be still placed into slice s to guarantee all observations are weighted equally. As OSIR has superior performance over SIR, we expect IOSIR will perform better than ISIR by a price of double calculation time.

We remark that SIR and ISIR can be used for both regression problems and classification problems. But since the concept of "adjacent slice" cannot be defined for categorical values (as is the case in classification problems), IOSIR can only be used for regression problems where the response variable is numeric.

4.5 Simulations

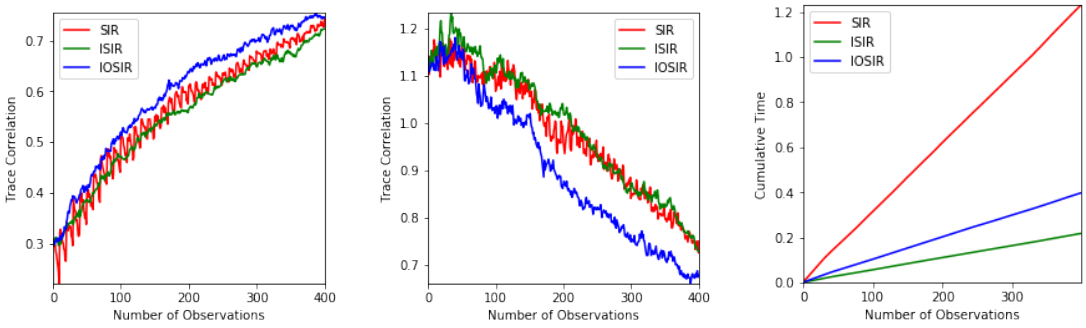
In this section, we will verify the effectiveness of ISIR and IOSIR with simulations on artificial and real-world data. Comparisons will be made between them and SIR.

4.5.1 Artificial data

In the simulations with artificial data, since we know the true model, we measure the performance by the accuracy of the estimated EDR space. We adopt the trace correlation $r(K) = \text{trace}(\mathbf{P_B}\mathbf{P}_{\widehat{\mathbf{B}}})/K$ used in [34], where $\mathbf{P_B}$ and $\mathbf{P}_{\widehat{\mathbf{B}}}$ are the projection operators onto the true EDR space $\mathbf{B}$ and the estimated EDR space $\widehat{\mathbf{B}}$, respectively, and the angle between $\mathbf{B}$ and $\hat{\mathbf{B}}$ [10, 101] as the criteria. We consider the model (2.5) from Chapter 2,

$$y = x_1(x_1 + x_2 + 1) + \epsilon,$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_p]^\top$ follow multivariate normal distribution, ϵ follows standard normal distribution and is independent of $\mathbf{x}$. It has $K = 2$ effective dimensions with $\boldsymbol{\beta}_1 = (1, 0, 0, \ldots, 0)^\top$ and $\boldsymbol{\beta}_2 = (0, 1, 0 \ldots, 0)^\top$. We conduct the simulation in $p = 10$ dimensional space and select the number of slices as $H = 10$. We give the algorithm a warm start with the initial guess of the EDR space obtained by applying SIR algorithm to a small data set of 40 observations. Then a total of 400 new observations will be fed to update the EDR space one by one. SIR, ISIR, and IOSIR are applied when each observation was fed in and we calculate their trace correlation and cumulative computation time. We repeat this process 100 times. The mean trace correlation for each of the three methods is reported in Figure 4.1(a), the mean angle is in Figure 4.1(b) and the mean cumulative time is in Figure 4.1(c). We see that ISIR performs quite similar to SIR and IOSIR slightly outperforms both ISIR and SIR. ISIR is much faster than SIR and IOSIR gains higher accuracy by sacrificing on computation time. This verifies the convergence and efficiency of ISIR and IOSIR.



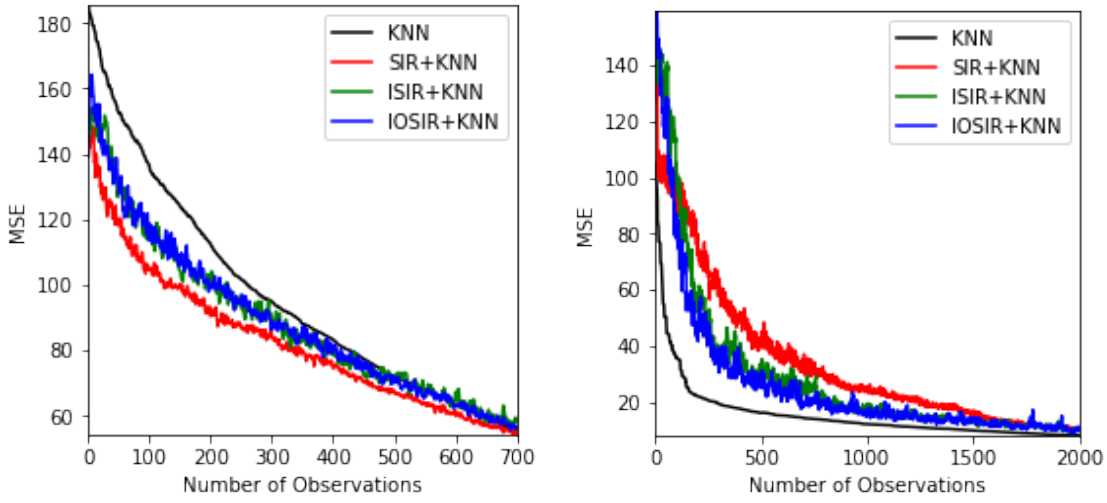
(a) (b) (c)

Figure 4.1. Performance and computational complexity of dimension reduction methods for artificial data generated from the model (2.5). (a) trace correlation; (b) angle; (c) cumulative computation time.

4.5.2 Real data applications

We validate the reliability of Incrmental SIR and Incremental OSIR on two data sets: Concrete Compressive Strength and Cpusmall (available on `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html`). There have been many proposed algorithms to increase the prediction accuracy on these data sets [44, 87, 114–116]. We do not intend to outperform those methods. Our goal is to compare the performance of supervised dimension reduction algorithms and verify the effectiveness and correctness of our incremental methods.

The Concrete Compressive Strength data has $p = 8$ predictors and 1030 samples. We use $H = 10$ and $K = 3$ to run SIR, ISIR, and IOSIR. We select 50 observations to warm start ISIR and IOSIR algorithms, then 700 observations are fed sequentially. The left 280 observations are left as test data. After each new observation is received we estimate the EDR space, project the available training set to the estimated EDR



(a) (b)

Figure 4.2. Mean square errors (MSE) for two real data applications: (a) Concrete Compressive Strength data and (b) Cpusmall data.

space, build a regression model using the k-nearest neighbor method, and compute the MSE on the test data set. This process is repeated 100 times and the average MSE was reported in Figure 4.2(a). For the Cpusmall data, which has $p = 12$ predictors and 8192 samples, we do the experiment with $H = 10$, $K = 3$, 50 observations to warm start ISIR and IOSIR, 2000 observation for sequential training, and 6142 observations for testing. The average MSE was plotted in Figure 4.2(b). The results indicate both ISIR and IOSIR are as effective as SIR.

4.6 Conclusions and discussions

We proposed two online learning approaches for supervised dimension reduction, namely, ISIR and IOSIR. They are motivated by standardizing the data and refor-

mulate the SIR algorithm to a PCA problem. However, data standardization is only used to motivate the algorithm while not explicitly calculated in the algorithms. We proposed to use Sherman Morrison formula to online update $\widehat{\mathbf{\Sigma}}^{-1}$ and some approximated calculation to circumvent explicit data standardization. This novel idea played a key role in our algorithm design. Both algorithms are shown effective and efficient. While IOSIR does not apply to classification problems, it is usually superior over ISIR in regression problems.

We remark that the purpose of ISIR and IOSIR is to keep the dimension reduction accuracy in the situation that a batch learning is not suitable. This is especially the case for streaming data where information update and system involving is necessary whenever new data becomes available. When the whole data is given and one only needs the EDR space from batch learning, ISIR or IOSIR is not necessarily more efficient than SIR because their complexity to run over the whole sample path is $O(p^2 n)$, comparable to the complexity $O(p^3 + p^2 n)$ of SIR.

There are two open problems worth further investigation. First, the need to store and use $\widehat{\mathbf{\Sigma}}^{-1}$ during the updating process is the main bottleneck for ISIR and IOSIR when the dimensionality of the data is ultrahigh. Second, for SIR and other batch dimension reduction methods, many methods have been proposed to determine the intrinsic dimension K; see e.g. [3, 5, 14, 70, 84, 94]. They depend on all p eigenvalues of the generalized eigen-decomposition problem and are impractical for incremental learning. We do not have obvious solutions to these problems at this moment and would like to leave them for future research.

4.7 Inference of the inverse covariance matrix update

Suppose $\mathbf{A}$ is an invertible square matrix, and $\mathbf{u}$, $\mathbf{v}$ are column vectors, and $1 + \mathbf{v}^\top \mathbf{A}^{-1}\mathbf{u} \neq n+1$, we have

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^\top) = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^\top\mathbf{A}^{-1}}{1 + \mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u}}. \tag{4.10}$$

In our case, compare Equation (4.7) with Equation (4.10), if we let $\mathbf{A} = \frac{n}{n+1}\widehat{\mathbf{\Sigma}}$, $\mathbf{u} = \frac{\sqrt{n}}{n+1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})$, and it is clear $\mathbf{v} = \mathbf{u}$. Thus,

$$\begin{aligned}
\widehat{\mathbf{\Sigma}'}^{-1} &= (\mathbf{A} + \mathbf{u}\mathbf{u}^\top)^{-1} \\
&= \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{u}^\top\mathbf{A}^{-1}}{1 + \mathbf{u}^\top\mathbf{A}^{-1}\mathbf{u}} \\
&= (\frac{n}{n+1}\widehat{\mathbf{\Sigma}})^{-1} \\
&\quad - \frac{(\frac{n}{n+1}\widehat{\mathbf{\Sigma}})^{-1}\frac{\sqrt{n}}{n+1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})\left(\frac{\sqrt{n}}{n+1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})\right)^\top(\frac{n}{n+1}\widehat{\mathbf{\Sigma}})^{-1}}{1 + \left(\frac{\sqrt{n}}{n+1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})\right)^\top(\frac{n}{n+1}\widehat{\mathbf{\Sigma}})^{-1}\frac{\sqrt{n}}{n+1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})} \\
&= \frac{n+1}{n} + \frac{\frac{n+1}{n}\widehat{\mathbf{\Sigma}}^{-1}\frac{n}{(n+1)^2}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})(\mathbf{x}_{n+1} - \bar{\mathbf{x}})^\top\frac{n+1}{n}\widehat{\mathbf{\Sigma}}^{-1}}{1 + \frac{n}{(n+1)^2}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})\frac{n+1}{n}\widehat{\mathbf{\Sigma}}^{-1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})} \\
&= \frac{n+1}{n}\widehat{\mathbf{\Sigma}}^{-1} - \frac{\frac{1}{n}\widehat{\mathbf{\Sigma}}^{-1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})(\mathbf{x}_{n+1} - \bar{\mathbf{x}})^\top\widehat{\mathbf{\Sigma}}^{-1}}{1 + \frac{1}{n+1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})^\top\widehat{\mathbf{\Sigma}}^{-1}(\mathbf{x}_{n+1} - \bar{\mathbf{x}})}.
\end{aligned}$$

CHAPTER 5

COVARIANCE-FREE INCREMENTAL SLICED INVERSE REGRESSION

5.1 Introduction

In Chapter 4, we have proposed the incremental versions of the SIR and OSIR methods. The update of the EDR space $\widehat{\mathbf{B}}$ involves with plenty of matrix multiplications and needs a time-consuming eigen-decomposition even the computational complexity has been reduced from $O(p^3)$ to $O((K+1)^3)$, $K < p$. In this chapter, we aim to develop incremental learning methods for SIR and OSIR with reduced matrix multiplications for computational efficiency. We also want to skip the computation of eigen-decomposition problem and adjust each EDR directions individually rather than rotate the whole EDR space $\widehat{\mathbf{B}}$. Again we hope to find inspirations from incremental PCA algorithms. This time we focus on covariance-free methods for their higher efficiency and simpler implementation.

5.2 Candid covariance-free incremental PCA

Several covariance-free incremental PCA algorithms [85, 86, 93] have been proposed, but they all have convergence problems [102]. The two most state-of-art covariance-free incremental PCA algorithms are the candid covariance-free Incremental PCA (CCIPCA) [102] and the largest-eigenvalue-theory for incremental PCA (LET-IPCA) [111]. Both methods can achieve good convergence speed and accuracy. The only difference is the eigenvectors got by CCIPCA are dependent, while LET-IPCA can get these directions independently. For a traditional IPCA algorithm,

it must observe an open number of observations and the number is larger than the dimension of the observed vectors, covariance-free IPCA methods can also skip this requirement. In this chapter, we adopt the ideas of CCIPCA to motivate our incremental SIR and OSIR algorithms.

Suppose we have a sequentially acquired online data stream: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots$. We can assume $\mathbf{E}[\mathbf{x}] = \mathbf{0}$ without loss of generality. For an eigenvalue problem

$$\mathbf{\Sigma u} = \lambda \mathbf{u},$$

if we replace $\mathbf{\Sigma}$ and $\mathbf{u}$ on the left with their sample versions and set $\mathbf{v} = \lambda \mathbf{u}$, we have

$$\mathbf{v} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i{}^{\top} \mathbf{u}.$$

After receiving a new observation $\mathbf{x_{n+1}}$, we wish to update it to

$$\mathbf{v}' = \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbf{x}_i \mathbf{x}_i{}^{\top} \mathbf{u}'.$$

If we use $\frac{\mathbf{v}}{\|\mathbf{v}\|}$ to estimate $\mathbf{u}'$, we can get the following incremental expression:

$$\begin{aligned} \mathbf{v}' &= \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbf{x}_i \mathbf{x}_i^{\top} \frac{\mathbf{v}}{\|\mathbf{v}\|} \\ &= \frac{1}{n+1} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^{\top} \frac{\mathbf{v}}{\|\mathbf{v}\|} + \frac{1}{n+1} \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\top} \frac{\mathbf{v}}{\|\mathbf{v}\|} \\ &= \frac{n}{n+1} \mathbf{v} + \frac{1}{n+1} \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\top} \frac{\mathbf{v}}{\|\mathbf{v}\|}. \end{aligned} \tag{5.1}$$

This is the updating formula for CCIPCA. Note that $\frac{n}{n+1}$ is the weight for the previous estimate and $\frac{1}{n+1}$ is the weight for the new observation. They indicates all observations in the data stream are equally weighted toward the calculation of principal components. To speed up the convergence, one may prefer to give a smaller weight to the early samples by changing the weights. This leads to the following updating formula

$$\mathbf{v}' = \frac{n-l}{n+1} \mathbf{v} + \frac{1+l}{n+1} \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\top} \frac{\mathbf{v}}{\|\mathbf{v}\|}, \tag{5.2}$$

where $l > 0$ is a re-weighting parameter and typically l ranges from 2 to 4 [102]. With the presence of l, a larger weight is given to new sample while the effort of old samples will fade out gradually. We can use Equation (5.2) as the incremental estimate of the first direction. When $\mathbf{v}$ converges, it is clear that $\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$.

Equations (5.1) and (5.2) only work for the first direction [120]. For the second direction, we can generate an "observation" in the complementary space of the first eigenvector by subtracting the projection of $\mathbf{x}_{n+1}$ onto the first direction from itself. The residual

$$\mathbf{x}_{n+1}^{(2)} = \mathbf{x}_{n+1}^{(1)} - \mathbf{x}_{n+1}^{(1)^\top} \frac{\mathbf{v}^{(1)\prime}}{\|\mathbf{v}^{(1)\prime}\|} \frac{\mathbf{v}^{(1)\prime}}{\|\mathbf{v}^{(1)\prime}\|}, \tag{5.3}$$

where $\mathbf{x}_{n+1}^{(1)} = \mathbf{x}_{n+1}$, $\mathbf{v}^{(1)\prime} = \mathbf{v}'$, can then be used to update the second direction by

$$\mathbf{v}^{(2)\prime} = \frac{n-l}{n+1}\mathbf{v}^{(2)} + \frac{1+l}{n+1}\mathbf{x}_{n+1}^{(2)}\mathbf{x}_{n+1}^{(2)^\top} \frac{\mathbf{v}^{(2)}}{\|\mathbf{v}^{(2)}\|}. \tag{5.4}$$

We can similarly update other directions. Note that the first K new observations will be used to initialize $\mathbf{v}^{(i)}, i = 1, 2, \ldots, K$. CCIPCA promises the convergence even if some eigenvalues are equal.

5.3 Covariance-free incremental SIR

To apply the idea of the CCIPCA and develop incremental algorithms for SIR and OSIR, we need to define new observations suitable for the EDR space or a space that can be transformed into EDR space. By [70, Theorem 3.1], the centered inverse regression curve $\mathbf{E}[\mathbf{x}|y] - \mathbf{E}[\mathbf{x}]$ is contained in the linear subspace spanned by $\mathbf{\Sigma}\boldsymbol{\beta}_k, k = 1, 2, \ldots, K$. We can consider $\mathbf{z} = \mathbf{E}[\mathbf{x}|y] - \mathbf{E}[\mathbf{x}]$ as the random variable of interest and the approximation of random drawers for $\mathbf{z}$ as observations in this subspace. In sample version, when a new observation $(\mathbf{x}_{n+1}, y_{n+1})$ comes, we find which slice it falls in by the corresponding distances from y_{n+1} to each slice center. Suppose the

distance from y_{n+1} to $\bar{y}_s$ is the shortest, then we can update the sample mean $\bar{\mathbf{x}}$ and slice mean $\mathbf{m}_s$ via

$$\bar{\mathbf{x}}' = \frac{n}{n+1}\bar{\mathbf{x}} + \frac{1}{n+1}\mathbf{x}_{n+1}$$

and

$$\mathbf{m}_s' = \frac{n_s}{n_s+1}\mathbf{m}_s + \frac{1}{n_s+1}\mathbf{x}_{n+1},$$

respectively. The new observation for subspace $\mathbf{\Sigma B}$ is computed as

$$\mathbf{z} = \mathbf{m}_s' - \bar{\mathbf{x}}'.$$

Now, following the idea of CCIPCA we propose to update the k-th direction of the subspace $\mathbf{\Sigma B}$ by

$$\mathbf{v}^{(k)\prime} = \frac{n-l}{n+1}\mathbf{v}^{(k)} + \frac{1+l}{n+1}\mathbf{z}^{(k)}\mathbf{z}^{(k)\top}\frac{\mathbf{v}^{(k)}}{\|\mathbf{v}^{(k)}\|}, \tag{5.5}$$

where $\mathbf{z}^{(1)} = \mathbf{z}$ and for $k \geq 1$

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - \mathbf{z}^{(k)\top}\frac{\mathbf{v}^{(k)\prime}}{\|\mathbf{v}^{(k)\prime}\|}\frac{\mathbf{v}^{(k)\prime}}{\|\mathbf{v}^{(k)\prime}\|}. \tag{5.6}$$

For the two weights $\frac{n-l}{n+1}$ and $\frac{1+l}{n+1}$, we can simply set $l = 0$ if $\frac{n-l}{n+1} \leq 0$. After the convergence of $\mathbf{V} = [\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(K)}]$, we can easily compute $\boldsymbol{H} = [\boldsymbol{\eta}^{(1)}, \boldsymbol{\eta}^{(2)}, \ldots, \boldsymbol{\eta}^{(K)}]$, where $\boldsymbol{\eta}^{(k)} = \frac{\mathbf{v}^{(k)}}{\|\mathbf{v}^{(k)}\|}$, which spans the subspace in which the centered inverse regression curve $\mathbf{E}[\mathbf{x}|y] - \mathbf{E}[\mathbf{x}]$ lies. Finally we can recover the EDR space by

$$\widehat{\mathbf{B}} = \widehat{\mathbf{\Sigma}}^{-1}\boldsymbol{H}, \tag{5.7}$$

where again we can use Sherman-Morrison formula to incrementally update $\widehat{\mathbf{\Sigma}}^{-1}$ incrementally. We call this method the covariance-free incremental SIR (CFISIR).

The primary computational cost of CFISIR results from the matrix multiplication, which has a complexity of $O(p^2)$. Note that Sherman-Morrison formula also circumvent the computational load of matrix inversion with a complexity of at least

$O(p^{2.373})$ [68] and the complexity of SIR is $O(p^3)$ due to the eigen-decomposition. In terms of storage, CFISIR only needs to keep matrices $\mathbf{V}$, $\mathbf{M} = [\mathbf{m_1}, \mathbf{m_2}, \ldots, \mathbf{m_k}]$, vectors $\bar{\mathbf{x}}$, $\bar{\mathbf{y}} = [\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_H,]$ and some scalars. For high dimensional data, the computational complexity and memory requirement will be reduced significantly.

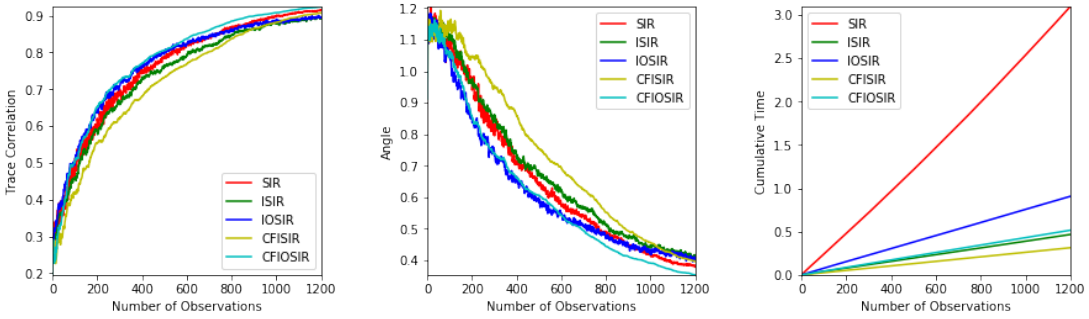
We note that, by following the same process of IOSIR in Chapter 4, we can propose the covariance-free incremental OSIR (CFIOSIR) method.

5.4 Simulations

In this section, we will prove the the accuracy and convergence of CFISIR and CFIOSIR on both artificial and real world data. We compare their performance with ISIR, IOSIR and SIR.

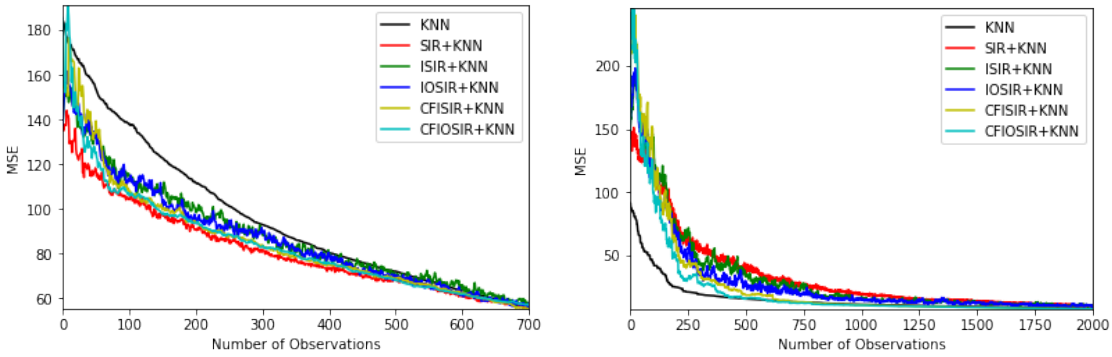
Our simulations use the same model and real data and follow exactly the same settings as in Chapter 4. For the artificial data we use both trace correlation and the angle between $\mathbf{B}$ and $\hat{\mathbf{B}}$ as the evaluation criteria. The experiment results are reported in Figure 5.1. They indicate the CFISIR and CFIOSIR have similar performance as other methods but are more computationally efficient.

For real data we use the KNN regression with $k = 5$ nearest neighbors to make prediction and the mean squared error (MSE) is used as the evaluation criterion. We repeat this process 100 times and report the average errors in Figure 5.2. Again we see that both CFISIR and CFIOSIR work as well as the original SIR and other incremental SIR algorithms.



(a) (b) (c)

Figure 5.1. Performance and computational complexity of dimension reduction methods for artificial data generated from the model (2.5). (a) trace correlation; (b) angle; (c) cumulative computation time.



(a) (b)

Figure 5.2. MSE of various methods on two real data sets. (a) Concrete; (b) Cpusmall.

5.5 Conclusions and Discussions

We proposed two incremental learning algorithms, CFISIR and CFIOSIR, for supervised dimension reduction. They are able to incrementally update the EDR space without the burden of the storage of data and the computational complexity of eigen-decomposition. Similar to ISIR and IOSIR, they are good candidates when a batch learning is not suitable, especially when data is acquired streamingly and it is impossible to store all the observations or a real-time system is required to respond or update instantly.

Again, similar to ISIR and IOSIR, determining the intrinsic dimensionality remains an open probem for CFISIR and CFIOSIR. We do not have obvious solutions to this problems at this moment and would like to leave it for future research.

CHAPTER 6

SUMMARY AND FUTURE WORK

6.1 Summary

Based on the different ways of receiving data and achieving the new model, machine learning techniques can be categorized as batch learning and incremental learning algorithms. We refined a classic dimension reduction algorithm in regression analysis, the sliced inverse regression (SIR), to proposed several new batch learning and incremental learning algorithms.

For batch learning, we aimed to improve the performance of SIR when the number of observations we have in hand is relatively limited. We introduced a slicing overlapping technique to code the information of the derivative of $\mathbf{E}[\mathbf{X}|y]$ and proposed a new algorithm called overlapping sliced inverse regression (OSIR). We applied two bootstrapping techniques, the bagging and an alternative bootstrapping method with extended Jacobian angles for simultaneous diagonalization, to "generate" more samples and "average" the corresponding estimated EDR spaces or covariance matrices in the group of generalized eigen-decomposition equations, which motivated the bagging OSIR and an alternative bootstrapping OSIR. Both overlapping and bootstrapping strategies are able to extract more information from limited observations and improve the accuracy of EDR space estimation.

For incremental learning, we aimed to update the existing EDR space when new observations are received as a streaming sequence and system update is required in an online manner. We proposed two versions of incremental SIR. The difference is their approaches to adjust the directions of the EDR space. ISIR needs a warm up

by applying SIR to a small bunch of observations, then extracts the information of the new coming observation to adjust EDR space as a whole. The warm up is not necessary for CFISIR, we can even choose the initial directions randomly, and the directions of EDR space are adjusted one by one, so we can also avoid the eigen-decomposition problem. We also applied the idea of overlapping into the incremental learning methods to propose IOSIR and CFIOSIR algorithms.

The effectiveness and efficiency of these new algorithms are justified by consistency analysis and simulation studies on both artificial and real world data.

6.2 Future work

In Chapter 3, an alternative way to implement bootstrapping OSIR is to run OSIR on bootstrapping samples and generate a set of projection matrices

$$\{\widehat{\mathbf{P}}^{*(t)}|t = 1, 2, \ldots, T\},$$

where $\widehat{\mathbf{P}}^{*(t)} = \widehat{\mathbf{B}}^{*(t)}\widehat{\mathbf{B}}^{*(t)^\top}$ is symmetric and $\widehat{\mathbf{B}}^{*(t)}$ is an estimated EDR space. After diagonalization, we can find the averaged projection matrix

$$\widehat{\mathbf{P}} = \mathbf{U}\bar{\mathbf{\Lambda}}'\mathbf{U}^\top,$$

where $\bar{\mathbf{\Lambda}}'$ is the average of all the diagonal matrices. We believe that the EDR space can be recovered from $\widehat{\mathbf{P}}$, further study will try to build this connection. There are a bunch of SIR-based algorithms, which are also worth investigating with the bootstrapping and overlapping techniques.

We had tried another version of ISIR method other than the version studied in Chapter 4. Recall that SIR method is associated to a generalized eigen-decomposition problem in Equation (1.5). It can be converted into a regular eigen-decomposition

problem by multiplying an inverse covariance matrix on both sides:

$$\mathbf{\Sigma^{-1}\Gamma B} = \mathbf{B\Lambda}.$$

Applying the idea of incremental PCA, an alternative approach to update EDR space is to solve the eigen-decompostion problem

$$[\mathbf{B}, \hat{\mathbf{v}}]^\top \mathbf{\Sigma'}^{-1} \mathbf{\Gamma'} [\mathbf{B}, \hat{\mathbf{v}}] \mathbf{R} = \mathbf{R\Lambda'}.$$

There is no estimate in this approach so that we preserve the accuracy when we update the EDR space. According to the theoretical analysis and simulation, it is also faster than the ISIR method we have proposed in Chapter 4. But a problem is that the matrix $[\mathbf{B}, \hat{\mathbf{v}}]^\top \mathbf{\Sigma'}^{-1} \mathbf{\Gamma'} [\mathbf{B}, \hat{\mathbf{v}}]$ is not symmetric, which causes a computational problem: the accumulation of computational inaccuracies will yield complex numbers. The simulation results show that the problem only occurs after the first two directions, it gives us a hope to refine this algorithm and overcome this problem by some fancier eigen-decomposition algorithm.

The incremental SIR methods we proposed in Chapter 4 and Chapter 5 can only deal with the new observations one by one. When a bunch of observations come, it is expected to extract useful information at once to update the model. It is necessary to investigate new incremental methods for this mini-batch learning setting.

The determination of the intrinsic dimensionality K in incremental SIR methods is another open problem for us. Most existing methods find K by using all the p eigenvalues of the generalized eigen-decomposition problem of SIR. They cannot be extended to the incremental learning setting because our methods only have $K + 1$ eigenvalues computed in ISIR and IOSIR, and even no eigenvalues computed in CFISIR and CFIOSIR. New techniques to tackle this problem should be developed in the future.

Due to the wide range of applications of SIR and other SIR-like algorithms, we believe it is worth applying our new algorithms for supervised dimension reduction to more real-world data in the future.

REFERENCES

[1] Edward I Altman, Giancarlo Marco, and Franco Varetto. Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the italian experience). *Journal of Banking & Finance*, 18(3):505–529, 1994.

[2] Anestis Antoniadis, Sophie Lambert-Lacroix, and Frédérique Leblanc. Effective dimension reduction methods for tumor classification using gene expression data. *Bioinformatics*, 19(5):563–570, 2003.

[3] Zhidong Bai and Xuming He. A chi-square test for dimensionality with non-Gaussian data. *Journal of Multivariate Analysis*, 88(1):109–117, 2004.

[4] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis - A brief tutorial. *Institute for Signal and Information Processing*, 18:1–8, 1998.

[5] M Pilar Barrios and Santiago Velilla. A bootstrap method for assessing the dimension of a general regression problem. *Statistics & Probability Letters*, 77(3):247–255, 2007.

[6] Claudia Becker and Roland Fried. Sliced inverse regression for high-dimensional time series. In *Exploratory Data Analysis in Empirical Research*, pages 3–11. Springer, 2003.

[7] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[8] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250. ACM, 2001.

[9] Christopher M Bishop, Markus Svensén, and Christopher KI Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[10] Åke Björck and Gene H Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

[11] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[12] David R Brillinger. Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, 86(414):333–333, 1991.

[13] Efstathia Bura. Using linear smoothers to assess the structural dimension of regressions. *Statistica Sinica*, 13(1):143–162, 2003.

[14] Efstathia Bura and R Dennis Cook. Extending sliced inverse regression: The weighted chi-squared test. *Journal of the American Statistical Association*, 96(455):996–1003, 2001.

[15] Jean-Francois Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.

[16] Miguel A Carreira-Perpinán. A review of dimension reduction techniques. *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, 9:1–69, 1997.

[17] Shivkumar Chandrasekaran, Bangalore S Manjunath, Yuan-Fang Wang, Jay Winkeler, and Henry Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59(5):321–332, 1997.

[18] Marie Chavent, Stéphane Girard, Vanessa Kuentz-Simonet, Benoit Liquet, Thi Mong Ngoc Nguyen, and Jérôme Saracco. A sliced inverse regression approach for data stream. *Computational Statistics*, 29(5):1129–1152, 2014.

[19] Marie Chavent, Vanessa Kuentz, Benoıt Liquet, and Jérôme Saracco. A sliced inverse regression approach for a stratified population. *Communications in Statistics - Theory and Methods*, 40(21):3857–3878, 2011.

[20] Alessandro Chiancone, Florence Forbes, and Stéphane Girard. Student sliced inverse regression. *Computational Statistics & Data Analysis*, 113:441–456, 2017.

[21] Alessandro Chiancone, Stéphane Girard, and Jocelyn Chanussot. Collaborative sliced inverse regression. *Communications in Statistics - Theory and Methods*, 46(12):6035–6053, 2017.

[22] Delin Chu, Li-Zhi Liao, Michael Kwok-Po Ng, and Xiaoyan Wang. Incremental linear discriminant analysis: a fast algorithm and comparisons. *IEEE Transactions on Neural Networks and Learning Systems*, 26(11):2716–2735, 2015.

[23] R Dennis Cook. Using dimension-reduction subspaces to identify important inputs in models of physical systems. In *Proceedings of the section on Physical and Engineering Sciences*, pages 18–25. American Statistical Association Alexandria, VA, 1994.

[24] R Dennis Cook. Testing predictor contributions in sufficient dimension reduction. *The Annals of Statistics*, 32(3):1062–1092, 2004.

[25] R Dennis Cook. *Regression Graphics: Ideas for Studying Regressions through Graphics*, volume 482. John Wiley & Sons, 2009.

[26] R Dennis Cook, Bing Li, and Francesca Chiaromonte. Dimension reduction in regression without matrix inversion. *Biometrika*, 94(3):569–584, 2007.

[27] R Dennis Cook and Sanford Weisberg. Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, 86(414):328–332, 1991.

[28] R Dennis Cook and Xin Zhang. Fused estimators of the central subspace in sufficient dimension reduction. *Journal of the American Statistical Association*, 109(506):815–827, 2014.

[29] Jian J Dai, Linh Lieu, and David Rocke. Dimension reduction for classification with gene expression microarray data. *Statistical Applications in Genetics and Molecular Biology*, 5(1), 2006.

[30] Shanshan Ding and R Dennis Cook. Tensor sliced inverse regression. *Journal of Multivariate Analysis*, 133:216–231, 2015.

[31] Yuexiao Dong and Liping Zhu. A note on sliced inverse regression with missing predictors. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(2):128–138, 2012.

[32] Laura Elnitski, Ross C Hardison, Jia Li, Shan Yang, Diana Kolbe, Pallavi Eswara, Michael J O'Connor, Scott Schwartz, Webb Miller, and Francesca Chiaromonte. Distinguishing regulatory DNA from neutral sites. *Genome Research*, 13(1):64–72, 2003.

[33] Randall L Eubank. *Nonparametric Regression and Spline Smoothing*. CRC press, 1999.

[34] Louis Ferré. Determining the dimension in sliced inverse regression and related methods. *Journal of the American Statistical Association*, 93(441):132–140, 1998.

[35] Louis Ferré and Anne-Françoise Yao. Functional sliced inverse regression analysis. *Statistics*, 37(6):475–488, 2003.

[36] Louis Ferré and Anne-Françoise Yao. Smoothed functional inverse regression. *Statistica Sinica*, 15:665–683, 2005.

[37] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Kernel dimensionality reduction for supervised learning. In *Advances in Neural Information Processing Systems*, pages 81–88, 2004.

[38] Ali Gannoun, Stéphane Girard, Christiane Guinot, and Jerôme Saracco. Sliced inverse regression in reference curves estimation. *Computational Statistics & Data Analysis*, 46(1):103–122, 2004.

[39] Ali Gannoun and Jérôme Saracco. An asymptotic theory for SIR_α method. *Statistica Sinica*, 13(2):297–310, 2003.

[40] Ursula Gather, Torsten Hilker, and Claudia Becker. A robustified version of sliced inverse regression. In *Statistics in Genetics and in the Environmental Sciences*, pages 147–157. Springer, 2001.

[41] Allen Gersho and Robert M Gray. *Vector Quantization and Signal Compression*, volume 159. Springer Science & Business Media, 2012.

[42] Youness Aliyari Ghassabeh, Abouzar Ghavami, and Hamid Abrishami Moghaddam. A new incremental face recognition system. *IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 335–340, 2007.

[43] Geof H. Givens and Jennifer A. Hoeting. *Computational Statistics*. New York: Wiley, 2nd edition, 2013.

[44] Bin Gu, Victor S Sheng, Zhijie Wang, Derek Ho, Said Osman, and Shuo Li. Incremental learning for ν-support vector regression. *Neural Networks*, 67:140–150, 2015.

[45] Peter M Hall, David A Marshall, and Ralph R Martin. Incremental eigenanalysis for classification. In *BMVC*, volume 98, pages 286–295, 1998.

[46] Peter M Hall, David A Marshall, and Ralph R Martin. Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1049, 2000.

[47] John HL Hansen, Abhishek Kumar, and Pongtep Angkititrakul. Environment mismatch compensation using average eigenspace-based methods for robust speech recognition. *International Journal of Speech Technology*, 17(4):353–364, 2014.

[48] Wolfgang Karl Härdle and Alexandre B. Tsybakov. Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, 86(414):333–335, 1991.

[49] David Harrison and Daniel L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

[50] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[51] Ping He, Kai-Tai Fang, and Cheng-Jian Xu. The classification tree combined with SIR and its applications to classification of mass spectra. *Journal of Data Science*, 1(4):425–445, 2003.

[52] Haileab Hilafu. Random sliced inverse regression. *Communications in Statistics-Simulation and Computation*, 46(5):3516–3526, 2017.

[53] Hideitsu Hino, Keigo Wakayama, and Noboru Murata. Entropy-based sliced inverse regression. *Computational Statistics & Data Analysis*, 67:105–114, 2013.

[54] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[55] Tailen Hsing and Raymond J Carroll. An asymptotic theory for sliced inverse regression. *The Annals of Statistics*, 20(2):1040–1061, 1992.

[56] Tailen Hsing and Haobo Ren. An RKHS formulation of the inverse regression dimension-reduction problem. *The Annals of Statistics*, 37(2):726–755, 2009.

[57] Peter J Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[58] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*, volume 46. John Wiley & Sons, 2004.

[59] Bo Jiang and Jun S Liu. Variable selection for general index models via sliced inverse regression. *The Annals of Statistics*, 42(5):1751–1786, 2014.

[60] Yu Wei Jiang, Ci-Ren and Jane-Ling Wang. Inverse regression for longitudinal data. *The Annals of Statistics*, 42(2):563–591, 2014.

[61] Ian Jolliffe. *Principal Component Analysis*. Springer, 2011.

[62] John T Kent. Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, 86(414):336–337, 1991.

[63] Tae-Kyun Kim, Shu-Fai Wong, Bjorn Stenger, Josef Kittler, and Roberto Cipolla. Incremental linear discriminant analysis using sufficient spanning set approximations. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[64] Teuvo Kohonen. *Self-organizing Maps*, volume 30. Springer Science & Business Media, 2012.

[65] Thomas Kötter. Asymptotic results for sliced inverse regression. *Acta Universitatis Lodziensis. Folia Oeconomica*, 141:73–82, 1997.

[66] Vanessa Kuentz, Benoît Liquet, and Jérôme Saracco. Bagging versions of sliced inverse regression. *Communications in Statistics - Theory and Methods*, 39(11):1985–1996, 2010.

[67] Martin HC Law and Anil K Jain. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):377–391, 2006.

[68] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 296–303. ACM, 2014.

[69] Bing Li and Yuexiao Dong. Dimension reduction for nonelliptically distributed predictors. *The Annals of Statistics*, 37(3):1272–1298, 2009.

[70] Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[71] Ker-Chau Li. Sliced inverse regression for dimension reduction: Rejoinder. *Journal of the American Statistical Association*, 86(414):337–342, 1991.

[72] Ker-Chau Li. On principal hessian directions for data visualization and dimension reduction: Another application of Stein's lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992.

[73] Lexin Li, R Dennis Cook, and Christopher J Nachtsheim. SIR^3: dimension reduction in the presence of linearly or nonlinearly related predictors. *Institute of Statistics Mimeo*, 2004.

[74] Lexin Li, R Dennis Cook, and Chih-Ling Tsai. Partial inverse regression. *Biometrika*, 94(3):615–625, 2007.

[75] Lexin Li and Hongzhe Li. Dimension reduction methods for microarrays with application to censored survival data. *Bioinformatics*, 20(18):3406–3412, 2004.

[76] Lexin Li and Christopher J Nachtsheim. Sparse sliced inverse regression. *Technometrics*, 48(4):503–510, 2006.

[77] Lexin Li and Xiangrong Yin. Sliced inverse regression with regularizations. *Biometrics*, 64(1):124–131, 2008.

[78] Mariëlle Linting, Jacqueline J Meulman, Patrick JF Groenen, and Anita J van der Koojj. Nonlinear principal components analysis: Introduction and application. *Psychological Methods*, 12(3):336, 2007.

[79] Heng-Hui Lue. Sliced inverse regression for multivariate response regression. *Journal of Statistical Planning and Inference*, 139(8):2656–2664, 2009.

[80] Kai Mao, Feng Liang, and Sayan Mukherjee. Supervised dimension reduction using bayesian mixture modeling. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 501–508, 2010.

[81] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural Networks*

for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (cat. no. 98th8468), pages 41–48. IEEE, 1999.

[82] Prasad A Naik, Michael R Hagerty, and Chih-Ling Tsai. A new dimension reduction approach for data-rich marketing environments: sliced inverse regression. *Journal of Marketing Research*, 37(1):88–101, 2000.

[83] Liqiang Ni, R Dennis Cook, and Chih-Ling Tsai. A note on shrinkage sliced inverse regression. *Biometrika*, 92(1):242–247, 2005.

[84] Guy Martial Nkiet. Consistent estimation of the dimensionality in sliced inverse regression. *Annals of the Institute of Statistical Mathematics*, 60(2):257–271, 2008.

[85] Erkki Oja. Subspace methods of pattern recognition. In *Pattern Recognition and Image Processing Series*, volume 6. Research Studies Press, 1983.

[86] Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106(1):69–84, 1985.

[87] Ahmet Öztaş, Murat Pala, Erdogan Özbay, Erdogan Kanca, Naci Caglar, and M Asghar Bhatti. Predicting the compressive strength and slump of high strength concrete using neural network. *Construction and Building Materials*, 20(9):769–775, 2006.

[88] Shaoning Pang, Seiichi Ozawa, and Nikola Kasabov. Incremental linear discriminant analysis for classification of data streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):905–914, 2005.

[89] Robi Polikar, Lalita Upda, Satish S Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, part C (Applications and Reviews)*, 31(4):497–508, 2001.

[90] Chuan-Xian Ren and Dao-Qing Dai. Incremental learning of bidirectional principal components for face recognition. *Pattern Recognition*, 43(1):318–330, 2010.

[91] Paul Rodriguez and Brendt Wohlberg. A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3414–3416. IEEE, 2014.

[92] Franz Rothlauf. Representations for genetic and evolutionary algorithms. In *Representations for Genetic and Evolutionary Algorithms*, pages 9–32. Springer, 2006.

[93] Terence D Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473, 1989.

[94] James R Schott. Determining the dimensionality in sliced inverse regression. *Journal of the American Statistical Association*, 89(425):141–148, 1994.

[95] Luca Scrucca. Model-based SIR for dimension reduction. *Computational Statistics & Data Analysis*, 55(11):3010–3026, 2011.

[96] C Messan Setodji and R Dennis Cook. K-means inverse regression. *Technometrics*, 46(4):421–429, 2004.

[97] Fengxi Song, David Zhang, Qinglong Chen, and Jingyu Yang. A novel supervised dimensionality reduction algorithm for online image recognition. In *Pacific-Rim Symposium on Image and Video Technology*, pages 198–207. Springer, 2006.

[98] Bruce Thompson. Factor analysis. *The Blackwell Encyclopedia of Sociology*, 2007.

[99] Jian-Gang Wang, Eric Sung, and Wei-Yun Yau. Incremental two-dimensional linear discriminant analysis with applications to face recognition. *Journal of Network and Computer Applications*, 33(3):314–322, 2010.

[100] Tao Wang, Xuerong Meggie Wen, and Lixing Zhu. Multiple-population shrinkage estimation via sliced inverse regression. *Statistics and Computing*, 27(1):103–114, 2017.

[101] Per Åke Wedin. On angles between subspaces of a finite dimensional inner product space. In *Matrix Pencils*, pages 263–285. Springer, 1983.

[102] Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang. Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):1034–1040, 2003.

[103] Han-Ming Wu. Kernel sliced inverse regression with applications to classification. *Journal of Computational and Graphical Statistics*, 17(3):590–610, 2008.

[104] Han-Ming Wu and Henry Horng-Shing Lu. Supervised motion segmentation by spatial-frequential analysis and dynamic sliced inverse regression. *Statistica Sinica*, 14(2):413–430, 2004.

[105] Qiang Wu, Justin Guinney, Mauro Maggioni, and Sayan Mukherjee. Learning gradients: Predictive models that infer geometry and statistical dependence. *Journal of Machine Learning Research*, 11(Aug):2175–2198, 2010.

[106] Qiang Wu, Feng Liang, and Sayan Mukherjee. Localized sliced inverse regression. *Journal of Computational and Graphical Statistics*, 19(4):843–860, 2010.

[107] Yingcun Xia, Howell Tong, Wai Keungxs Li, and Li-Xing Zhu. An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):363–410, 2002.

[108] Wang Xiaoyan. *Incremental and Regularized Linear Discriminant Analysis*. PhD thesis, National University OF Singapore, 21 Lower Kent Ridge Rd, Singapore 119077, 8 2012.

[109] Xiao-Lin Xu, Chuan-Xian Ren, Ran-Chao Wu, and Hong Yan. Sliced inverse regression with adaptive spectral sparsity for dimension reduction. *IEEE Transactions on Cybernetics*, 47(3):759–771, 2017.

[110] Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Towards incremental and large scale face recognition. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–6. IEEE, 2011.

[111] Shuicheng Yan and Xiaoou Tang. Largest-eigenvalue-theory for incremental principal component analysis. In *IEEE International Conference on Image Processing 2005*, volume 1, pages I–1181. IEEE, 2005.

[112] Jieping Ye, Qi Li, Hui Xiong, Haesun Park, Ravi Janardan, and Vipin Kumar. IDR/QR: An incremental dimension reduction algorithm via QR decomposi-

tion. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1208–1222, 2005.

[113] Zhishen Ye and Jie Yang. Sliced inverse moment regression using weighted chi-squared tests for dimension reduction. *Journal of Statistical Planning and Inference*, 140(11):3121–3131, 2010.

[114] I-Cheng Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

[115] I-Cheng Yeh. Design of high-performance concrete mixture using neural networks and nonlinear programming. *Journal of Computing in Civil Engineering*, 13(1):36–42, 1999.

[116] I-Cheng Yeh. Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites*, 29(6):474–480, 2007.

[117] Xiangrong Yin and R Dennis Cook. Estimating central subspaces via inverse third moments. *Biometrika*, 90(1):113–125, 2003.

[118] Zhou Yu, Yuexiao Dong, and Liping Zhu. Distance weighted inverse regression for dimension reduction and variable selection. preprint, 2017.

[119] Ting Zhang, Wenhua Ye, and Yicai Shan. Application of sliced inverse regression with fuzzy clustering for thermal error modeling of CNC machine tool. *The International Journal of Advanced Manufacturing Technology*, 85(9-12):2761–2771, 2016.

[120] Yilu Zhang and Juyang Weng. Convergence analysis of complementary candid incremental principal component analysis. *Michigan State University*, 2001.

[121] Haitao Zhao and Pong Chi Yuen. Incremental linear discriminant analysis for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(1):210–221, 2008.

[122] Haitao Zhao, Pong Chi Yuen, and James T Kwok. A novel incremental principal component analysis and its application for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(4):873–886, 2006.

[123] Li-Ping Zhu, Li-Xing Zhu, and Zheng-Hui Feng. Dimension reduction in regressions through cumulative slicing estimation. *Journal of the American Statistical Association*, 105(492):1455–1466, 2010.

[124] Li-Xing Zhu and Kai-Tai Fang. Asymptotics for kernel estimate of sliced inverse regression. *The Annals of Statistics*, 24(3):1053–1068, 1996.

[125] Li-Xing Zhu and Kai W Ng. Asymptotics of sliced inverse regression. *Statistica Sinica*, 5(2):727–736, 1995.

[126] Lixing Zhu, Baiqi Miao, and Heng Peng. On sliced inverse regression with high-dimensional covariates. *Journal of the American Statistical Association*, 101(474):630–643, 2006.