Journal of Machine Learning Research 21 (2020) 1-25 Submitted 8/18; Revised 1/20; Published 1/20

# Online Sufficient Dimension Reduction Through Sliced Inverse Regression

**Zhanrui Cai** ZXC55@PSU.EDU
**Runze Li** RZLI@PSU.EDU
*Department of Statistics*
*The Pennsylvania State University*
*University Park, PA 16802, USA*

**Liping Zhu** ZHU.LIPING@RUC.EDU.CN
*Center for Applied Statistics*
*Institute of Statistics and Big Data*
*Renmin University of China*
*Beijing 100872, China*

**Editor:** Jie Peng

## Abstract

Sliced inverse regression is an effective paradigm that achieves the goal of dimension reduction through replacing high dimensional covariates with a small number of linear combinations. It does not impose parametric assumptions on the dependence structure. More importantly, such a reduction of dimension is sufficient in that it does not cause loss of information. In this paper, we adapt the stationary sliced inverse regression to cope with the rapidly changing environments. We propose to implement sliced inverse regression in an online fashion. This online learner consists of two steps. In the first step we construct an online estimate for the kernel matrix; in the second step we propose two online algorithms, one is motivated by the perturbation method and the other is originated from the gradient descent optimization, to perform online singular value decomposition. The theoretical properties of this online learner are established. We demonstrate the numerical performance of this online learner through simulations and real world applications. All numerical studies confirm that this online learner performs as well as the batch learner.

**Keywords:** Dimension reduction, online learning, perturbation, singular value decomposition, sliced inverse regression, gradient descent.

## 1. Introduction

With the advances in sensor technology and computer hardware, increasing amounts of high dimensional data are being rapidly collected in many scientific and social studies. Suppose $Y \in \mathbb{R}^1$ is a response variable and $\mathbf{x} = (X_1, \ldots, X_p)^\top \in \mathbb{R}^{p\times 1}$ are the associated covariates. To deal with the high dimensionality of $\mathbf{x}$, one of the popular ideas is to replace the high dimensional $\mathbf{x}$ with a small number of linear combinations, say $(\mathbf{B}^\top\mathbf{x})$, where $\mathbf{B}$ is a $p \times K$ matrix. By the very purpose of dimension reduction, $K$ is far less than $p$. In many applications, $K = 1, 2$ or at most 3, depending on the complexity of the underlying data structure. Sufficient dimension reduction (Li, 1991; Cook, 2009) is a paradigm that combines the idea of linear dimension reduction with the concept of statistical sufficiency.

To be precise, it seeks for a matrix $\mathbf{B} \in \mathbb{R}^{p\times K}$ such that

$$Y \perp\!\!\!\perp \mathbf{x} \mid (\mathbf{B}^\top \mathbf{x}), \tag{1}$$

where $\perp\!\!\!\perp$ stands for statistical independence. The sufficiency is described as the fact that, in model (1), $(Y \mid \mathbf{x})$ and $(Y \mid \mathbf{B}^\top \mathbf{x})$ follow identical distributions. In other words, using $(\mathbf{B}^\top \mathbf{x})$ to predict $Y$ has the same power of using $\mathbf{x}$. Therefore, replacing $\mathbf{x}$ with $(\mathbf{B}^\top \mathbf{x})$ is sufficient to characterize the dependence of $(Y \mid \mathbf{x})$, and this replacement does not cause any loss of information. In addition, this reduction does not impose parametric assumptions on the dependence structure. These two appealing properties make sufficient dimension reduction very attractive in high dimensional data analysis because we are often lack of information on the underlying dependence structure. We are hesitated to specify the dependence structure because a mis-specified model may lead to severe bias in estimation.

In model (1), the basis matrix $\mathbf{B}$ is not identifiable. Instead of estimating $\mathbf{B}$, the ultimate goal of sufficient dimension reduction is to find the central subspace, denoted by $\mathcal{S}_{Y|\mathbf{x}}$ and defined as the column space of $\mathbf{B}$ with minimal dimension. With a slight abuse of notation, we write $\mathcal{S}_{Y|\mathbf{x}} = \text{span}(\mathbf{B})$ and denote $K$ to be the structural dimension of $\mathcal{S}_{Y|\mathbf{x}}$.

Since the seminal work of sliced inverse regression (Li, 1991), many sufficient dimension reduction methods have been developed to recover $\mathcal{S}_{Y|\mathbf{x}}$. This includes, but not limited to, sliced average variance estimation (Cook and Weisberg, 1991), directional regression (Li and Wang, 2007), cumulative slicing estimation (Zhu et al., 2010), semi-parametric approaches (Ma and Zhu, 2012), and minimum average variance estimation (Xia et al., 2002). See Ma and Zhu (2013) and Li (2018) for a comprehensive review on recent advances in the area of sufficient dimension reduction. All these methods focus on the setting of batch learning.

In the big data era, how to perform dimension reduction on a stochastic data stream is important and yet rarely touched in the literature. Our main interest is to adapt existing sufficient dimension reduction methods to cope with the rapidly changing environments where the observations arrive sequentially as $\{(\mathbf{x}_t, Y_t), t = 1, \ldots\}$ in a data stream. We advocate using sufficient dimension reduction because it remain valid even when the dependence structures vary arbitrarily in the process of generating a data stream. The online learner, which significantly reduces the computational complexity of the batch learner with a small sacrifice of accuracy, is one of the most popular choices for solving large-scale problems (Bousquet and Bottou, 2008). In this article we propose to implement sliced inverse regression (Li, 1991) in an online fashion. This online learner consists of two steps. In the first step an online estimate for the kernel matrix of sliced inverse regression is constructed; in the second step two online algorithms are proposed, one is motivated by the perturbation method and the other is originated from the gradient descent optimization, to perform online singular value decomposition. The perturbation method and the gradient descent optimization, in addition to the randomized algorithms (Warmuth and Kuzmin, 2008; Boutsidis et al., 2015; Nie et al., 2016), are widely used in the online principal component analysis. See Li et al. (2000), Hegde et al. (2006), Arora et al. (2013) and Yang and Xu (2015) for their respective applications of the perturbation method and the gradient descent optimization. Despite its popularity, the theoretical properties of the perturbation method has yet been studied for online singular value decomposition. In this article, we establish the consistency of the perturbation method based on the quasi-martingale theory (Fisk, 1965). The theoretical convergence of the gradient descent optimization has been analyzed

only for independent data (Oja and Karhunen, 1985; Balsubramani et al., 2013; Shamir, 2016). In our proposed two-step online learning algorithm, the first step gives a sequence of solutions which act as the instances for the second step. Because the observations up to the $t$-th step are also used in the $(t+1)$-th step, the instances given by the first step are highly correlated, violating the independence assumption which is typically required to analyze the convergence properties of the online singular value decomposition problems (Arora et al., 2012; Mitliagkas et al., 2013; Arora et al., 2013; Shamir, 2016). In this paper, we accommodate highly correlated instances and show the strong convergence of the online gradient descent optimization.

To accommodate data streams arriving sequentially by blocks, Chavent et al. (2014) adapted sliced inverse regression to a block-wise version. They proposed to divide the entire data set into many small data blocks, and perform sliced inverse regression on each of these small blocks. Aggregating all the estimates yields a final estimate. Chavent et al. (2014)'s methodology is more like a divide-and-conquer strategy, which is common for parallel distributed computing but uncommon in the online learning community. An important requirement in Chavent et al. (2014) is that the data streams must arrive in blocks with sufficiently many observations. In the present context, we assume that one observation arrives after another in the data streams, which appears more challenging and is usually the focus of online learning. Another relevant work is Bercu et al. (2015). They studied the recursive sliced inverse regression with only two slices. In this case, the kernel matrix of sliced inverse regression has at most one nonzero eigenvalue and the principal eigenvector has an explicit form. The singular value decomposition step is thus completely avoided. In the present context, we consider sliced inverse regression with a general number of slices. In this case, the principal eigenvectors do not have explicit forms and singular value decomposition step is essential. In this sense, our proposed two-step online sliced inverse regression procedure is much more challenging and general than existing works.

This paper is organized as follows. In Section 2 we propose a two-step online procedure for sliced inverse regression. Its theoretical properties are also established. In Section 3, we demonstrate the numerical performance of our proposed procedure through simulations and several benchmark datasets available from the UCI machine learning repository. We provide some concluding remarks in Section 4. All proofs are relegated to the Appendix.

## 2. Online Sliced Inverse Regression

We first give a brief review of the classic sliced inverse regression (Li, 1991). Denote $\mathbf{\Sigma} = \mathrm{cov}(\mathbf{x})$, the covariance matrix of $\mathbf{x}$. When $Y$ is continuous, the slicing procedure partitions the support of $Y$ into $H$ slices, $I_1, \ldots, I_H$, where $I_h = (q_{h-1}, q_h]$, and $-\infty = q_0 < q_1 < \cdots < q_H = \infty$. Although Li (1991) suggested to set $q_h$ to be the $(h/H) \times 100\%$-th quantile of $Y$, the cutting points $q_h$s do not have to be the quantiles of $Y$. Let $p_h \stackrel{\mathrm{def}}{=} \Pr(Y \in I_h)$ and $\mathbf{m}_h^0 \stackrel{\mathrm{def}}{=} \mathbf{\Sigma}^{-1}\mathbb{E}\{(\mathbf{x} - \mathbb{E}\mathbf{x}) \mid Y \in I_h\}$. If $Y$ is categorical or discrete taking $H$ distinct values, say, $1, \ldots, H$, we can simply replace $\{Y \in I_h\}$ in $p_h$ and $\mathbf{m}_h^0$ with $\{Y = h\}$, for $h = 1, \ldots, H$. The kernel matrix of the classic sliced inverse regression is defined by

$$\mathbf{M}^0 \stackrel{\mathrm{def}}{=} \sum_{h=1}^{H} p_h \mathbf{m}_h^0 {\mathbf{m}_h^0}^\top.$$

Li (1991) showed that, under mild conditions on $\mathbf{x}$,

$$\mathbf{m}_h^0 \in \mathcal{S}_{Y|\mathbf{x}}, \text{ for } h = 1, \ldots, H, \tag{2}$$

which implies that the first $K$ principal eigenvectors of $\mathbf{M}^0$ span $\mathcal{S}_{Y|\mathbf{x}}$ and the $(p-K)$ smallest eigenvalues of $\mathbf{M}^0$ are identically zero. At the sample level, we replace $q_h$s with the sample quantiles $\widehat{q}_h$s and estimate $\mathbf{\Sigma}$, $p_h$, $\mathbf{m}_h^0$ and $\mathbf{M}^0$ with their moment estimates $\widehat{\mathbf{\Sigma}}$, $\widehat{p_h}$, $\widehat{\mathbf{m}}_h^0$ and $\widehat{\mathbf{M}}^0$, respectively. The space spanned by the first $K$ principal eigenvectors of $\widehat{\mathbf{M}}^0$ yields a consistent estimate of $\mathcal{S}_{Y|\mathbf{x}}$. Li (1991) also claimed that, the efficiency of sliced inverse regression is very insensitive to the number of slices $H$.

In this paper, we are concerned with the situation where the observations arrive sequentially as $\{(\mathbf{x}_t, Y_t), t = 1, \ldots\}$ in a data stream. To perform online dimension reduction, we suggest to modify sliced inverse regression slightly when the response is a continuous variable. First, the slices $I_h$s must be pre-specified. This is trivial when $Y$ is categorical or discrete taking finite number of values. When $Y$ is continuous, its possible ranges can be roughly decided in real world problems. We partition the ranges of $Y$ into $H$ slices, $I_1, \ldots, I_H$, where $I_h = (q_{h-1}, q_h]$, and $q_h$s are pre-specified cutting points satisfying $-\infty = q_0 < q_1 < \cdots < q_H = \infty$. If the slices were not pre-specified and we insisted to use the quantiles of $Y$ in the slicing estimation, the sample quantiles would vary from time to time as the data stream $\{(\mathbf{x}_t, Y_t), t = 1, \ldots\}$ is evolving, resulting in unnecessary complexities for an online algorithm. Define

$$\mathbf{m}_h \overset{\text{def}}{=} \mathbf{\Sigma}^{-1}\mathbb{E}\{(\mathbf{x} - \mathbb{E}\mathbf{x})\mathbb{1}(Y \in I_h)\} \text{ and } \mathbf{M} \overset{\text{def}}{=} \sum_{h=1}^{H} \mathbf{m}_h \mathbf{m}_h{}^{\top}.$$

The fact that $\mathbf{m}_h^0 = \mathbf{m}_h / p_h$ and $p_h$ is a scalar, together with (2), immediately implies

$$\mathbf{m}_h \in \mathcal{S}_{Y|\mathbf{x}}, \text{ for } h = 1, \ldots, H, \tag{3}$$

and accordingly, the first $K$ principal eigenvectors of $\mathbf{M}$ span $\mathcal{S}_{Y|\mathbf{x}}$. The sliced inverse regression remains valid for an arbitrary slicing procedure as long as $-\infty = q_0 < q_1 < \cdots < q_H = \infty$. Therefore, we are allowed to use $\mathbf{M}$ instead of $\mathbf{M}^0$ to recover $\mathcal{S}_{Y|\mathbf{x}}$. In practice, we can use a small batch of data to advise a series of cutting points $q_1, \ldots, q_{H-1}$. This is our second suggestion of modification. The strategy of using $\mathbf{m}_h$ in lieu of $\mathbf{m}_h^0$ is very useful. Because the cutting points $q_h$s are pre-specified, it may happen that only a small portion of observations falls into the $h$-th interval $I_h = (q_{h-1}, q_h]$, leading to a trivially small $p_h$ and possibly very unstable estimates of $\mathbf{m}_h^0$ and $\mathbf{M}^0$. We advocate using the kernel matrix $\mathbf{M}$ instead of $\mathbf{M}^0$ in sliced inverse regression, which ensures to yield a stable estimate of $\mathcal{S}_{Y|\mathbf{x}}$. The second modification indeed leads to a variation of cumulative slicing estimation proposed by Zhu et al. (2010). These modifications facilitate the online implementation of sliced inverse regression significantly.

### 2.1 Online Update for the Kernel Matrix

In the present context we assume the observations arrive sequentially as $\{(\mathbf{x}_t, Y_t), t = 1, \ldots\}$ in a data stream. We use the first $t$ observations, $\{(\mathbf{x}_i, Y_i), i = 1, \ldots, t\}$, to estimate $\mathbf{m}_h$ and

$\mathbf{M}$, which yields $\widehat{\mathbf{m}}_{t,h}$ and $\widehat{\mathbf{M}}_t$, respectively. In parallel to the definition of $\mathbf{M}$, we define

$$\widehat{\mathbf{M}}_t \stackrel{\text{def}}{=} \sum_{h=1}^{H} \widehat{\mathbf{m}}_{t,h}\widehat{\mathbf{m}}_{t,h}^\top = (\widehat{\mathbf{m}}_{t,1}, \ldots, \widehat{\mathbf{m}}_{t,H})(\widehat{\mathbf{m}}_{t,1}, \ldots, \widehat{\mathbf{m}}_{t,H})^\top, \tag{4}$$

where

$$\widehat{\mathbf{\Sigma}}_t \stackrel{\text{def}}{=} t^{-1}\sum_{i=1}^{t}(\mathbf{x}_i - \overline{\mathbf{x}}_t)(\mathbf{x}_i - \overline{\mathbf{x}}_t)^\top, \quad \overline{\mathbf{x}}_t \stackrel{\text{def}}{=} t^{-1}\sum_{i=1}^{t}\mathbf{x}_i,$$
$$\widehat{\mathbf{m}}_{t,h} \stackrel{\text{def}}{=} \widehat{\mathbf{\Sigma}}_t^{-1}\left\{t^{-1}\sum_{i=1}^{t}(\mathbf{x}_i - \overline{\mathbf{x}}_t)\mathbb{1}(Y_i \in I_h)\right\}.$$

By definition, online updating $\widehat{\mathbf{M}}_{t+1}$ from $\widehat{\mathbf{M}}_t$ amounts to updating $\widehat{\mathbf{m}}_{t+1,h}$ from $\widehat{\mathbf{m}}_{t,h}$, for $h = 1, \ldots, H$, when the $(t+1)$-th observation $(\mathbf{x}_{t+1}, Y_{t+1})$ arrives.

Next we discuss how to update $\widehat{\mathbf{m}}_{t+1,h}$ from $\widehat{\mathbf{m}}_{t,h}$ using an online least squares approach. We augment $\mathbf{x}$ by adding a row of ones, $\widetilde{\mathbf{x}} = (1, \mathbf{x}^\top)^\top \in \mathbb{R}^{(p+1)\times 1}$. Similarly, we define $\widetilde{\mathbf{x}}_t = (1, \mathbf{x}_t^\top)^\top$, $t = 1, \ldots$. Adding a row of ones allows us to include an intercept in the least squares approach. We notice that, $\mathbf{m}_h$ is precisely the slope vector by regressing $\mathbb{1}(Y \in I_h)$ on $\widetilde{\mathbf{x}}$ linearly, and at the sample level, $\widehat{\mathbf{m}}_{t,h}$ is the least squares estimate of the slope vector by regressing $\{\mathbb{1}(Y_1 \in I_h), \ldots, \mathbb{1}(Y_t \in I_h)\}^\top$ onto $(\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_t)^\top$ linearly. For each $h$, the augmented design matrix $(\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_t)^\top$ are exactly the same. Therefore, we can simultaneously update $(\widehat{\mathbf{m}}_{t+1,1}, \ldots, \widehat{\mathbf{m}}_{t+1,H})$ from $(\widehat{\mathbf{m}}_{t,1}, \ldots, \widehat{\mathbf{m}}_{t,H})$. To be precise, we form $\widetilde{\mathbf{y}}_t = \{\mathbb{1}(Y_t \in I_1), \ldots, \mathbb{1}(Y_t \in I_H)\}^\top \in \mathbb{R}^{H\times 1}$. We further define $\mathbb{0}_{p\times 1}$ to be a $p$-vector of zeros, and $\mathbf{I}_{p\times p}$ to be the $p \times p$ identity matrix. Write $\widetilde{\mathbf{I}} \stackrel{\text{def}}{=} (\mathbb{0}_{p\times 1}, \mathbf{I}_{p\times p})$ and

$$\mathbf{A}_t = \sum_{i=1}^{t}\widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^\top \in \mathbb{R}^{(p+1)\times(p+1)}.$$

After simple algebra,

$$\mathbf{A}_{t+1}^{-1} = \mathbf{A}_t^{-1} - \frac{\mathbf{A}_t^{-1}\widetilde{\mathbf{x}}_{t+1}\widetilde{\mathbf{x}}_{t+1}^\top\mathbf{A}_t^{-1}}{1 + \widetilde{\mathbf{x}}_{t+1}^\top\mathbf{A}_t^{-1}\widetilde{\mathbf{x}}_{t+1}}. \text{ Define } \mathbf{C}_{t+1} \stackrel{\text{def}}{=} \widetilde{\mathbf{I}}_{p\times(p+1)}\left(\mathbf{A}_t^{-1} - \frac{\mathbf{A}_t^{-1}\widetilde{\mathbf{x}}_{t+1}\widetilde{\mathbf{x}}_{t+1}^\top\mathbf{A}_t^{-1}}{1 + \widetilde{\mathbf{x}}_{t+1}^\top\mathbf{A}_t^{-1}\widetilde{\mathbf{x}}_{t+1}}\right).$$

It follows immediately that

$$\begin{aligned}(\widehat{\mathbf{m}}_{t,1}, \ldots, \widehat{\mathbf{m}}_{t,H}) &= \widetilde{\mathbf{I}}_{p\times(p+1)}\mathbf{A}_t^{-1}\left(\sum_{i=1}^{t}\widetilde{\mathbf{x}}_i\widetilde{\mathbf{y}}_i^\top\right), \text{ and consequently,} \\ (\widehat{\mathbf{m}}_{t+1,1}, \ldots, \widehat{\mathbf{m}}_{t+1,H}) &= \mathbf{C}_{t+1}\left(\sum_{i=1}^{t}\widetilde{\mathbf{x}}_i\widetilde{\mathbf{y}}_i^\top + \widetilde{\mathbf{x}}_{t+1}\widetilde{\mathbf{y}}_{t+1}^\top\right). \end{aligned} \tag{5}$$

The notations $\widehat{\mathbf{m}}_{t,h}$s and $\widehat{\mathbf{m}}_{t+1,h}$s are defined in an obvious manner. Left multiplying $\widetilde{\mathbf{I}}_{p\times(p+1)}$ allows us to extract the slope vectors, or equivalently, to exclude the intercepts of the least squares estimate. The display in (5) describes how we can update $(\widehat{\mathbf{m}}_{t+1,1}, \ldots, \widehat{\mathbf{m}}_{t+1,H})$

from $\mathbf{A}_t^{-1}$ and $(\widehat{\mathbf{m}}_{t,1}, \ldots, \widehat{\mathbf{m}}_{t,H})$ precisely, at the $(t+1)$-th iteration. By definition in (4), updating $\widehat{\mathbf{M}}_t$ at the $(t+1)$-th iteration is straightforward. To be precise,

$$\widehat{\mathbf{M}}_{t+1} \stackrel{\text{def}}{=} \sum_{h=1}^{H} \widehat{\mathbf{m}}_{t+1,h} \widehat{\mathbf{m}}_{t+1,h}^{\top}.$$

It is remarkable here that, online updating $\widehat{\mathbf{m}}_{t,h}$ through (5) guarantees that

$$\begin{aligned} \widehat{\mathbf{m}}_{t+1,h} &= \widehat{\mathbf{\Sigma}}_{t+1}^{-1} \left\{ (t+1)^{-1} \sum_{i=1}^{t+1} (\mathbf{x}_i - \overline{\mathbf{x}}_{t+1}) \mathbb{1}(Y_i \in I_h) \right\}, \text{ where} \qquad (6) \\ \widehat{\mathbf{\Sigma}}_{t+1} &= (t+1)^{-1} \sum_{i=1}^{t+1} (\mathbf{x}_i - \overline{\mathbf{x}}_{t+1})(\mathbf{x}_i - \overline{\mathbf{x}}_{t+1})^{\top}, \text{ and } \overline{\mathbf{x}}_{+1} = (t+1)^{-1} \sum_{i=1}^{t+1} \mathbf{x}_i. \end{aligned}$$

Directly updating $\widehat{\mathbf{m}}_{t,h}$ through (6) involves the inversion of $\widehat{\mathbf{\Sigma}}_{t+1}$, which is typically regarded as computationally prohibitive. By contrast, online updating $\widehat{\mathbf{m}}_{t,h}$ through (5) takes the advantage of updating $\widehat{\mathbf{\Sigma}}_{t+1}^{-1}$ from $\widehat{\mathbf{\Sigma}}_t^{-1}$, which relaxes the computational complexity substantially. Therefore, we advocate updating $\widehat{\mathbf{m}}_{t,h}$ through (5) rather than (6), although both (5) and (6) yield an identical solution.

By invoking similar arguments of Zhu and Ng (1995), we can easily prove that both $\widehat{\mathbf{m}}_{t+1,h}$ and $\widehat{\mathbf{M}}_{t+1}$ are root-$t$-consistent. In the present context, the slices $I_h$s are pre-specified with fixed cutting points, $\widehat{\mathbf{m}}_{t+1,h}$ and $\widehat{\mathbf{M}}_{t+1}$ are usual moment estimates. By assuming that the fourth moment exists, or equivalently, $E(\mathbf{x}^{\top}\mathbf{x})^2 < \infty$, it is thus easier to show that both $\widehat{\mathbf{m}}_{t+1,h}$ and $\widehat{\mathbf{M}}_{t+1}$ are root-$t$-consistent. In other words, $\widehat{\mathbf{m}}_{t+1,h} - \mathbf{m}_h = O_p(t^{-1/2})$, for $h = 1, \ldots, H$, and $\widehat{\mathbf{M}}_{t+1} - \mathbf{M} = O_p(t^{-1/2})$. Details are omitted from the present context.

### 2.2 Online Singular Value Decomposition

Estimating $\mathcal{S}_{Y|\mathbf{x}}$ at the $t$-th iteration amounts to seeking for the $K$ principal singular vectors of $\widehat{\mathbf{M}}_t$. In this section we propose two online algorithms, one is motivated by the perturbation method and the other is originated from the gradient descent optimization, to perform an online singular value decomposition.

We first introduce the online singular value decomposition using the idea of perturbation. The following lemma is a key to online learning and also a classic result in perturbation theory of linear operators (Sibson, 1979; Kato, 2013).

**Lemma 1** *Let $\mathbf{D} \in \mathbb{R}^{p \times p}$ be a symmetric matrix and $(\lambda_j, \mathbf{v}_j)$ be the eigen-pairs of $\mathbf{D}$, $j = 1, \ldots, p$. Assume $|\lambda_1| > \cdots > |\lambda_d| > \lambda_{d+1} = \cdots = \lambda_p = 0$. Let $\varepsilon$ be a very small constant and $\mathbf{G}$ be a symmetric matrix. Denote the first order perturbation $\mathbf{D}(\varepsilon) = \mathbf{D} + \varepsilon\mathbf{G} + O(\varepsilon^2)$ and the eigen-pairs of $\mathbf{D}(\varepsilon)$ by $\{\lambda_j(\varepsilon), \mathbf{v}_j(\varepsilon)\}$. Then*

$$\begin{aligned} \lambda_j(\varepsilon) &= \lambda_j + \varepsilon(\mathbf{v}_j^{\top}\mathbf{G}\mathbf{v}_j) + O(\varepsilon^2), \text{ and} \\ \mathbf{v}_j(\varepsilon) &= \mathbf{v}_j + \varepsilon(\lambda_j \mathbf{I}_{p\times p} - \mathbf{D})^{+}\mathbf{G}\mathbf{v}_j + O(\varepsilon^2), \ j = 1, \ldots, d, \end{aligned}$$

*where $(\lambda_j \mathbf{I}_{p\times p} - \mathbf{D})^{+}$ stands for the Moore-Penrose pseudo-inverse of $(\lambda_j \mathbf{I}_{p\times p} - \mathbf{D})$ and $\mathbf{I}_{p\times p}$ stands for the $p \times p$ identity matrix.*

To put this lemma into online learning, we define

$$\widehat{\mathbf{\Gamma}}_t \stackrel{\text{def}}{=} t^{-1}\sum_{i=1}^{t}\widehat{\mathbf{M}}_t. \text{ It follows that } \widehat{\mathbf{\Gamma}}_{t+1} = \widehat{\mathbf{\Gamma}}_t - (t+1)^{-1}(\widehat{\mathbf{\Gamma}}_t - \widehat{\mathbf{M}}_{t+1}). \tag{7}$$

We apply Lemma 1 by defining $\mathbf{D} = \widehat{\mathbf{\Gamma}}_t$, $\mathbf{G} = \widehat{\mathbf{\Gamma}}_t - \widehat{\mathbf{M}}_{t+1}$ and $\varepsilon = -(t+1)^{-1}$. Let $(\widehat{\lambda}_{t,j}, \widehat{\boldsymbol{\beta}}_{t,j})$ be the eigen-pairs of $\widehat{\mathbf{\Gamma}}_t$, $j = 1, \cdots, p$. The algorithm for online singular value decomposition with the permutation method is described in Algorithm 1.

**Algorithm 1** Online sliced inverse regression via the perturbation method

1. Initialize $\widehat{\mathbf{M}}_1$ and $\widehat{\mathbf{\Gamma}}_1$ to obtain $(\widehat{\lambda}_{1,j}, \widehat{\boldsymbol{\beta}}_{1,j})$, $j = 1, \cdots, K$, with a small sample.
2. As the data stream is being collected,

2a) online update for the kernel matrix: update $\widehat{\mathbf{M}}_{t+1}$ from $\widehat{\mathbf{M}}_t$ through (5) to obtain

$$\widehat{\mathbf{M}}_{t+1} = \sum_{h=1}^{H}\widehat{\mathbf{m}}_{t+1,h}\widehat{\mathbf{m}}_{t+1,h}^{\top}, \tag{8}$$

and update $\widehat{\mathbf{\Gamma}}_{t+1}$ from $\widehat{\mathbf{\Gamma}}_t$ through (7);

2b) online singular value decomposition: update the eigen-pairs of $\widehat{\mathbf{\Gamma}}_{t+1}$ through

$$\widehat{\lambda}_{t+1,j} = \widehat{\lambda}_{t,j} - (t+1)^{-1}\widehat{\boldsymbol{\beta}}_{t,j}^{\top}(\widehat{\mathbf{\Gamma}}_t - \widehat{\mathbf{M}}_{t+1})\widehat{\boldsymbol{\beta}}_{t,j}, \text{ and} \tag{9}$$

$$\widehat{\boldsymbol{\beta}}_{t+1,j} = \widehat{\boldsymbol{\beta}}_{t,j} - (t+1)^{-1}(\widehat{\lambda}_{t,j}\mathbf{I}_{p\times p} - \widehat{\mathbf{\Gamma}}_t)^{+}(\widehat{\mathbf{\Gamma}}_t - \mathbf{M}_{t+1})\widehat{\boldsymbol{\beta}}_{t,j}. \tag{10}$$

3. Output $\widehat{\mathbf{B}}_{t+1} = (\widehat{\boldsymbol{\beta}}_{t+1,1}, \cdots, \widehat{\boldsymbol{\beta}}_{t+1,K})$.

Note that $\widehat{\mathbf{B}}_{t+1}$ in the third step of Algorithm 1 may not be orthogonal due to small computation errors. Thus an orthonormalization step may be needed to stabilize the numerical performance, i.e., $\widehat{\mathbf{B}}_{t+1} = \mathcal{P}_{\text{orth}}(\widehat{\boldsymbol{\beta}}_{t+1,1}, \cdots, \widehat{\boldsymbol{\beta}}_{t+1,K})$, where $\mathcal{P}_{\text{orth}}()$ is an operator orthonormalizing matrix columns and can be realized by the Gram-Schmidt procedure. We remark here that this orthonormalization procedure is not necessary for all $t$.

The perturbation method is widely used in the online singular value decomposition. See, for example, Li et al. (2000) and Hegde et al. (2006). In the present context, we prove the strong convergence of our proposed online algorithm based on the quasi-martingale theory (Fisk, 1965). We first present the following regularity conditions.

(C1) The smallest eigenvalue of $\mathbf{\Sigma}$ is bounded away from zero.

(C2) The observations $\{(\mathbf{x}_t, Y_t), t = 1, \ldots\}$ are independent and identically distributed.

(C3) The nonzero eigenvalues of $\mathbf{M}$ are all distinct.

(C4) $E(\mathbf{x}|\mathbf{B}^{\top}\mathbf{x})$ is a linear function of the $K$-dimensional random vector $\mathbf{B}^{\top}\mathbf{x}$.

These conditions are mild and widely used in the literature. Condition (C1) requires that the covariance matrix is well-behaved. One can refer to Xiao (2010) and Mairal et al. (2010) for discussions of condition (C2) used in the literature of online learning. We impose condition (C3) to ensure identifiability of the corresponding eigenvectors. The identifiability conditional plays an important role in the perturbation theory. See, for example, Champagne (1994) and Hegde et al. (2006). (C4) is the linearity condition commonly assumed in the dimension reduction literature, see Li (1991) for example.

**Theorem 2** *Under Conditions (C1)-(C4), the eigen-pair* $(\widehat{\lambda}_{t,j}, \widehat{\boldsymbol{\beta}}_{t,j})$ *of* $\widehat{\boldsymbol{\Gamma}}_t$ *generated by Algorithm 1 converges almost surely as* $t \to \infty$*, for* $j = 1, \ldots, K$*.*

Bottou (1998) pointed out that online learning algorithms without almost surely convergence property can be misled by specific improbable examples, presented a general framework of the online learning algorithms and proved the almost sure convergence using stochastic approximation theory. Mairal et al. (2010) proved the almost surely convergence for online dictionary learning. Tarres and Yao (2014) and Lei et al. (2017) discussed the almost surely convergence of the online learning algorithms in reproducing kernel Hilbert spaces. Theorem 2 guarantees that the online perturbation algorithm converges almost surely. This ensures the resulting estimate is well defined, although updating the eigenvectors involves calculating Moore-Penrose pseudo-inverse repetitively, which significantly increases computational complexity.

In binary classification problems, a natural choice of $K$ is one because the kernel matrix $\mathbf{M}$ is known to have at most rank one. In general, however, $K$ is often unknown and has to be decided in an online and data-driven fashion. We discuss this issue in what follows. In Section 2.1, we stated that $\widehat{\mathbf{M}}_{t+1} - \mathbf{M} = O_p(t^{-1/2})$. The root-$t$-consistency of $\widehat{\mathbf{M}}_{t+1}$ ensures that, there exists a symmetric random matrix $\mathbf{G}$ such that $\widehat{\mathbf{M}}_{t+1} = \mathbf{M} + t^{-1/2}\mathbf{G} + o_p(t^{-1/2})$. This, together with the perturbation theory in Lemma 1, yields that, $\widehat{\lambda}_{t+1,j} = \lambda_j + t^{-1/2}\boldsymbol{\beta}_j^\top \mathbf{G}\boldsymbol{\beta}_j + o_p(t^{-1/2})$. Following Zhu et al. (2010), we define a BIC type criterion as follows:

$$\widehat{K}_t = \underset{1 \le k \le p}{\operatorname{argmax}}\ D_t(k)$$

where

$$D_t(k) = \sum_{j=1}^{k} \widehat{\lambda}_{t,j}^2 \Big/ \sum_{j=1}^{p} \widehat{\lambda}_{t,j}^2 - C_t k(k+1)/(2t).$$

By invoking the root-$t$-consistency of $\widehat{\lambda}_{t,j}$, Zhu et al. (2010) proved that $\widehat{K}_t$ converges in probability to $K$, as $t \to \infty$, as long as $t^{-1}C_t \to 0$ and $C_t \to \infty$. Deciding an optimal selection of $C_t$ is not straightforward. Throughout the present context we choose $C_t = t^{1/2}$, which performs quite well in our numerical studies. In practice, one may also use a small batch of data to estimate $K$, if $\widehat{K}_t$ does not have to be updated online.

Next we introduce an alternative gradient descent optimization for online singular value decomposition. The gradient descent optimization has also been widely used for online analysis of independent data streams. See, for example, Arora et al. (2012), Arora et al.

(2013) and Yang and Xu (2015). Its popularity owes to low computational complexity, low memory requirement and excellent performance in practice.

The rationale goes as follows. We notice that, seeking for the $K$ principal eigenvectors of $\widehat{\mathbf{M}}_t$ amounts to maximizing $\mathrm{tr}(\widehat{\mathbf{M}}_t\mathbf{B}\mathbf{B}^\top)$ over the manifold $\mathbf{B}^\top\mathbf{B} = \mathbf{I}_{K\times K}$, where tr() is the trace operator. The first order derivative of the objective function is $2\widehat{\mathbf{M}}_t\mathbf{B}$, which inspires the proposal of the following online gradient descent optimization:

$$\widehat{\mathbf{B}}_{t+1} = \mathcal{P}_{\mathrm{orth}}(\widehat{\mathbf{B}}_t + \gamma_{t+1}\widehat{\mathbf{M}}_t\widehat{\mathbf{B}}_t), \tag{11}$$

where $\gamma_{t+1}$ is a small step-size. This online gradient descent optimization is described in Algorithm 2.

**Algorithm 2** online sliced inverse regression via the gradient descent optimization

1. Initialize $\widehat{\mathbf{M}}_1$ to obtain $\widehat{\mathbf{B}}_1$ with a small sample.
2. As the data stream is being collected,
   - online update for the kernel matrix: update $\widehat{\mathbf{M}}_t$ from $\widehat{\mathbf{M}}_{t-1}$ through (8),
   - online singular value decomposition: update the eigenspace of $\widehat{\mathbf{M}}_t$ through (11).

The theoretical convergence of the gradient descent optimization has been extensively analyzed for independent instances (Oja and Karhunen, 1985; Balsubramani et al., 2013; Shamir, 2016). In (11) of our proposed online algorithm, $\widehat{\mathbf{M}}_{t+1}$ used at the $(t+1)$-th iteration and $\widehat{\mathbf{M}}_t$ used at the $t$-th iteration are highly correlated because they differ only in a single observation. The high correlation issue complicates the theoretical analysis dramatically. In Theorem 3, we accommodate this high correlation issue and show the almost sure convergence of the online gradient descent optimization.

To ensure the convergence of the gradient descent algorithm, $\gamma_t$ must satisfy the usual Robbins-Monro condition:

$$\sum_{t\geq 1}\gamma_t^2 < \infty \text{ and } \sum_{t\geq 1}\gamma_t = \infty.$$

In the present context, we make the following assumption to satisfy the Robbins-Monro condition and also simplify the proofs.

(C5) Let $\gamma_t = Ct^{-1}$, $t = 1, 2, \ldots$, where $C$ is some constant.

**Theorem 3** *Under Conditions (C2)-(C5), the column space of* $\widehat{\mathbf{B}}_t = (\widehat{\boldsymbol{\beta}}_{t,1}, \ldots, \widehat{\boldsymbol{\beta}}_{t,K})$ *converges almost surely to the column space of* $\mathbf{M}$*, as* $t \to \infty$.

The column space of $\mathbf{M}$, defined as the space spanned by the eigenvectors associated with nonzero eigenvalues, coincides with the central subspace $\mathcal{S}_{Y|\mathbf{X}}$, as demonstrated in (3). Theorem 2 shows that the proposed online algorithm for the eigen-pair of $\widehat{\mathbf{\Gamma}}_t$ is almost sure convergence. Theorem 3 provides us even stronger results since it shows that $\widehat{\mathbf{B}}_t$ is an asymptotically consistent basis matrix of $\mathcal{S}_{Y|\mathbf{X}}$. It is challenging in establishing

the asymptotical consistence due to strong dependence between $\widehat{\mathbf{M}}_t$ and $\widehat{\mathbf{M}}_{t+1}$. The strong convergence of online principal component analysis (PCA) has been analyzed in the literature (Oja and Karhunen, 1985; Balsubramani et al., 2013; Arora et al., 2012, 2013; Shamir, 2016). Note that the contribution of the $t$-th data points to online covariance matrix is additive, while this is not true for the $\widehat{\mathbf{M}}_t$ any more. This makes theoretical analysis of the online singular value decomposition (SVD) in SIR is much more complicated than the online PCA. In Theorem 3, we tackle this technical difficulty and show the strong convergence of the $\widehat{\mathbf{B}}_t$.

In the first step, the computational cost of online updating the kernel matrix is of order $O(p^2H)$, which does not depend on $t$. Next we compare the computational complexity of two online algorithms for singular value decomposition in the second step. The computational complexity of updating (8) is of order $O(p^2H)$. In addition, the computational cost of updating the eigen-pair $(\widehat{\lambda}_{t+1,j}, \widehat{\boldsymbol{\beta}}_{t+1,j})$ is dominated by that of (10). Therefore, we analyze (10) only in what follows. In (10), both $(\mathbf{I}_{p\times p} - \widehat{\mathbf{\Gamma}}_t)^+$ and $(\widehat{\mathbf{\Gamma}}_t - \mathbf{M}_{t+1})$ are $p \times p$ matrices, the computational complexity of their multiplication is of order $O(p^3)$, for each given $j$. Therefore, the total complexity of Algorithm 1 is of order $O(p^2H + p^3K)$. By contrast, in (11), $\widehat{\mathbf{M}}_t$ is a $p \times p$ matrix and $\widehat{\mathbf{B}}_t$ is a $p \times K$ matrix, the computational complexity of their multiplication is of order $O(p^2K)$. In addition, the orthonormalization step requires an additional computational cost of order $O(pK^2)$, which is apparently dominated by $O(p^2K)$. Therefore, the total complexity of Algorithm 2 is of order $O(p^2H+p^2K)$. The above analysis is formally stated in Proposition 4.

**Proposition 4** *The total computational complexity of Algorithm 1 is of order $O(p^2H + p^3K)$, and that of Algorithm 2 is of order $O(p^2H + p^2K)$.*

We remark here that, a major disadvantage of Algorithm 1 is that it requires to calculate $(\mathbf{I}_{p\times p} - \widehat{\mathbf{\Gamma}}_t)^+$ and $(\mathbf{I}_{p\times p} - \widehat{\mathbf{\Gamma}}_t)^+(\widehat{\mathbf{\Gamma}}_t - \mathbf{M}_{t+1})$, both of which have the computational cost of order $O(p^3)$. However, the computational complexity of both algorithms does not depend on $t$, which shows an obvious advantage of online learning.

### 2.3 An Extension

In this section, we generalize our proposed online learning to sliced average variance estimation (Cook and Weisberg, 1991). The classic slicing procedure partitions the range of $Y$ into $H$ slices, $I_h = (q_{h-1}, q_h]$, $h = 1, \ldots, H$. When the slices are given, the kernel matrix of the classic sliced average variance estimation is defined by

$$\mathbf{M}^1 \overset{\text{def}}{=} \sum_{h=1}^{H} p_h \mathbf{M}_h \mathbf{M}_h{}^\top.$$

where $\mathbf{M}_h \overset{\text{def}}{=} \mathbf{I}_{p\times p} - \mathbf{\Sigma}^{-1}\text{var}(\mathbf{x} \mid Y \in I_h)$. To facilitate implementing sliced average variance estimation in an online fashion, we propose to modify the kernel matrix slightly. To be precise, we define $\mathbf{\Lambda}_h \overset{\text{def}}{=} p_h\mathbf{I}_{p\times p} - \mathbf{\Sigma}^{-1}\text{var}\{\mathbf{x}\mathbb{1}(Y \in I_h)\}$, and

$$\mathbf{\Lambda} \overset{\text{def}}{=} \sum_{h=1}^{H} \mathbf{\Lambda}_h \mathbf{\Lambda}_h^\top.$$

Following Cook and Weisberg (1991), we can show that $\mathbf{\Lambda}_h \in \mathcal{S}_{Y|\mathbf{x}}$ under mild conditions of $\mathbf{x}$. Therefore, the eigenvectors associated with the non-zeros eigenvalues of $\mathbf{\Lambda}$ spans $\mathcal{S}_{Y|\mathbf{x}}$.

We suppose the observations, $\{(\mathbf{x}_t, Y_t), t = 1, \ldots, \}$, arrive sequentially in a data stream. In parallel to the online sliced inverse regression, the online sliced average variance estimation consists of two steps. In the first step, we require to update the kernel matrix $\mathbf{\Lambda}$ in an online fashion. This amounts to updating $\mathbf{\Lambda}_h$ sequentially. By definition,

$$\mathbf{\Sigma}^{-1}\mathrm{var}\left\{\mathbf{x}\mathbb{1}(Y \in I_h)\right\} = \mathbf{\Sigma}^{-1}\mathbb{E}\big\{\mathbf{x}\mathbf{x}^\top\mathbb{1}(Y \in I_h)\big\} - \mathbf{\Sigma}^{-1}\mathbb{E}\big\{\mathbf{x}\mathbb{1}(Y \in I_h)\big\}\mathbb{E}\big\{\mathbf{x}^\top\mathbb{1}(Y \in I_h)\big\}.$$

In addition to $p_h = \mathbb{E}\big\{\mathbb{1}(Y \in I_h)\big\}$, there are three quantities in the above display, $\mathbf{Q}_h \stackrel{\text{def}}{=} \mathbf{\Sigma}^{-1}\mathbb{E}\big\{\mathbf{x}\mathbf{x}^\top\mathbb{1}(Y \in I_h)\big\}$, $\mathbf{m}_h \stackrel{\text{def}}{=} \mathbf{\Sigma}^{-1}\mathbb{E}\big\{\mathbf{x}\mathbb{1}(Y \in I_h)\big\}$ and $\mathbf{u}_h \stackrel{\text{def}}{=} \mathbb{E}\big\{\mathbf{x}\mathbb{1}(Y \in I_h)\big\}$, that must be updated in an online fashion. We remark here that, updating $p_h$ and $\mathbf{u}_h$ is very similar, in that both are of the form of expectations and the inversion of $\mathbf{\Sigma}$, $\mathbf{\Sigma}^{-1}$, is not required. In addition, both $\mathbf{Q}_h$ and $\mathbf{m}_h$ can be updated with an online least squares approach.

We discuss online estimates of $\mathbf{Q}_h$ and $\mathbf{m}_h$ first. We first notice that both are the slope vectors of the least squares regression with different responses. This allows us to regress $\{\mathbf{x}_1\mathbb{1}(Y_1 \in I_h), \ldots, \mathbf{x}_t\mathbb{1}(Y_t \in I_h)\}^\top$ and $\{\mathbb{1}(Y_1 \in I_h), \ldots, \mathbb{1}(Y_t \in I_h)\}^\top$ onto $(\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_t)^\top$, where $\widetilde{\mathbf{x}} = (1, \mathbf{x}^\top)^\top \in \mathbb{R}^{(p+1)\times 1}$. The online least squares yields that

$$\begin{aligned}
\widehat{\mathbf{Q}}_{t+1,h} &\stackrel{\text{def}}{=} \mathbf{D}_{t+1}\left\{\sum_{i=1}^{t}\widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^\top\mathbb{1}(Y_i \in I_h) + \widetilde{\mathbf{x}}_{t+1}\widetilde{\mathbf{x}}_{t+1}^\top\mathbb{1}(Y_{t+1} \in I_h)\right\},\\
\widehat{\mathbf{m}}_{t+1,h} &\stackrel{\text{def}}{=} \mathbf{D}_{t+1}\left\{\sum_{i=1}^{t}\widetilde{\mathbf{x}}_i\mathbb{1}(Y_i \in I_h) + \widetilde{\mathbf{x}}_{t+1}\mathbb{1}(Y_{t+1} \in I_h)\right\},
\end{aligned}$$

where

$$\mathbf{D}_{t+1} \stackrel{\text{def}}{=} \widetilde{\mathbf{I}}_{p\times(p+1)}\left(\mathbf{A}_t^{-1} - \frac{\mathbf{A}_t^{-1}\widetilde{\mathbf{x}}_{t+1}\widetilde{\mathbf{x}}_{t+1}^\top\mathbf{A}_t^{-1}}{1 + \widetilde{\mathbf{x}}_{t+1}^\top\mathbf{A}_t^{-1}\widetilde{\mathbf{x}}_{t+1}}\right).$$

The online estimates of $\mathbf{u}_h$ and $p_h$ are given, respectively, by

$$\widehat{\mathbf{u}}_{t+1,h} \stackrel{\text{def}}{=} (t+1)^{-1}\big\{t\widehat{\mathbf{u}}_{t,h} + \mathbf{x}_{t+1}\mathbb{1}(Y_{t+1} \in I_h)\big\}, \quad \text{and} \quad \widehat{p}_{t+1,h} \stackrel{\text{def}}{=} (t+1)^{-1}\big\{t\widehat{p}_{t,h} + \mathbb{1}(Y_{t+1} \in I_h)\big\}.$$

With the above online estimates $\widehat{p}_{t+1,h}$, $\widehat{\mathbf{Q}}_{t+1,h}$, $\widehat{\mathbf{m}}_{t+1,h}$ and $\widehat{\mathbf{u}}_{t+1,h}$, we can easily construct an online estimate of $\mathbf{\Lambda}$. To be precise,

$$\widehat{\mathbf{\Lambda}}_{t+1} \stackrel{\text{def}}{=} \sum_{h=1}^{H}\widehat{\mathbf{\Lambda}}_{t+1,h}\widehat{\mathbf{\Lambda}}_{t+1,h}^\top, \text{ where } \widehat{\mathbf{\Lambda}}_{t+1,h} \stackrel{\text{def}}{=} \widehat{p}_{t+1,h}\mathbf{I}_{p\times p} - \big(\widehat{\mathbf{Q}}_{t+1,h} - \widehat{\mathbf{m}}_{t+1,h}\widehat{\mathbf{u}}_{t+1,h}^\top\big).$$

In the second step, we perform an online singular value decomposition on $\widehat{\mathbf{\Lambda}}_{t+1}$. Both the perturbation method and the gradient descent optimization proposed in Section 2.2 can be readily used, though it is required to replace $\widehat{\mathbf{M}}_{t+1}$ with $\widehat{\mathbf{\Lambda}}_{t+1}$ in both algorithms.

## 3. Numerical Validation

### 3.1 Simulation

In this section we evaluate the performance of our proposal through simulations. Throughout we consider the following three models.

$$Y = (\boldsymbol{\beta}_1^\top \mathbf{x}) + \varepsilon, \tag{12}$$
$$Y = (\boldsymbol{\beta}_2^\top \mathbf{x})^3 + \varepsilon, \tag{13}$$
$$Y = (\boldsymbol{\beta}_3^\top \mathbf{x})/\{1 + (\boldsymbol{\beta}_4^\top \mathbf{x} + 1)^2\} + 0.2\varepsilon. \tag{14}$$

In these three models, $\boldsymbol{\beta}_1 = (1, 1, 0, 0, \ldots, 0)^\top \in \mathbb{R}^{20\times 1}$, $\boldsymbol{\beta}_2 = (0, 0, 1, 0, \ldots, 0)^\top \in \mathbb{R}^{20\times 1}$, $\boldsymbol{\beta}_3 = (1, 0, 0, 0, \ldots, 0)^\top \in \mathbb{R}^{10\times 1}$ and $\boldsymbol{\beta}_4 = (0, 1, 0, 0, \ldots, 0)^\top \in \mathbb{R}^{10\times 1}$. The covariate vector $\mathbf{x}$ is drawn from multivariate standard normal distribution, and $\varepsilon$ is standard normal.

We compare the performance of the following competitors.

(M1) Online sliced inverse regression via the perturbation method.

(M2) Online sliced inverse regression via the gradient descent optimization.

The above proposals correspond to Algorithms 1 and 2 in Section 2.2, respectively.

(M3) Sliced inverse regression via batch learning. This corresponds to the classic sliced inverse regression proposed by Li (1991), and serves as a benchmark for comparison.

(M4) Sliced inverse regression via block-wise learning. Chavent et al. (2014) suggested this proposal to deal with data streams arriving in blocks.

Li (1991) stated that sliced inverse regression is insensitive to the number of slices. To implement sliced inverse regression (M1)-(M3) when the response variable is continuous, we simply set the slice number $H = 5$ throughout our numerical studies. To implement the blockwise sliced inverse regression (M4), we fix the block size to be 10.

(M5) Online principal component analysis via the perturbation method.

(M6) Online principal component analysis via the gradient descent optimization.

**Estimation Accuracy:** We first compare the estimation accuracy of the above competitors. Let $\mathbf{B}$ be an orthonormal basis of $\mathcal{S}_{Y|\mathbf{x}}$, which reduces to $\boldsymbol{\beta}_1/\|\boldsymbol{\beta}_1\|$ in model (12), $\boldsymbol{\beta}_2$ in model (13), and $(\boldsymbol{\beta}_3, \boldsymbol{\beta}_4)$ in model (14). Suppose $\widehat{\mathbf{B}}$ is an estimated orthonormal basis matrix. Ye and Weiss (2003) proposed to measure the distance between $\mathbf{B}$ and $\widehat{\mathbf{B}}$ through

$$d(\mathbf{B}, \widehat{\mathbf{B}}) \stackrel{\text{def}}{=} 1 - \left|\det(\mathbf{B}^\top \widehat{\mathbf{B}})\right| \tag{15}$$

where $\det(\cdot)$ stands for the determinant operator. It ranges from 0 and 1, with a smaller value indicating a better performance. Thus it is reasonable to use $d(\mathbf{B}, \widehat{\mathbf{B}})$ to measure the estimation accuracy of $\widehat{\mathbf{B}}$. We report the averaged distance based on 100 replications. The simulation results are summarized in Table 1. The classic sliced inverse regression (M3) serves as a benchmark. It is not surprising to see that all the supervised learners (M1)-(M4) perform better as the sample size $t$ increases from 1000 to 10000, and these

supervised learners are significantly superior to the unsupervised ones. The online sliced inverse regressions (M1) and (M2) perform much better than their competitors (M4)-(M6). In addition, (M2) is comparable to (M1) in the one-dimensional models (12) and (13), and is however much better in the two-dimensional model (14).

Algorithm 1 also gives an online estimate of the eigenvalues. This allows us to estimate the structural dimension $K$ of $\mathcal{S}_{Y|\mathbf{x}}$. We apply the BIC type criterion suggested in Section 2.2 to estimate $K$. Throughout all three models, the BIC type criterion yields an accurate estimate of $K$. The empirical probabilities of observing $\widehat{K} = 1$ in models (12)-(13) and $\widehat{K} = 2$ in model (14) are exactly one in our simulations.

Table 1: The averages of the distance $d(\mathbf{B}, \widehat{\mathbf{B}})$ based on 100 replications, where (M1) and (M2) stand for the respective online sliced inverse regression via the perturbation method and the gradient descent optimization, (M3) corresponds to the classic sliced inverse regression, and (M4) stands for the sliced inverse regression via block-wise learning, (M5) and (M6) stand for the respective online principal component analysis via the perturbation method and the gradient descent optimization.

| Models | sample size | (M1) | (M2) | (M3) | (M4) | (M5) | (M6) |
|---|---|---|---|---|---|---|---|
| (12) | 1000 | 0.0276 | 0.0996 | 0.0102 | 0.2256 | 0.8240 | 0.8598 |
| | 5000 | 0.0059 | 0.0196 | 0.0023 | 0.1415 | 0.8746 | 0.8493 |
| | 10000 | 0.0035 | 0.0112 | 0.0014 | 0.1431 | 0.8521 | 0.8491 |
| (13) | 1000 | 0.1476 | 0.2299 | 0.0320 | 0.5767 | 0.8168 | 0.8590 |
| | 5000 | 0.0422 | 0.0684 | 0.0105 | 0.5364 | 0.8558 | 0.8763 |
| | 10000 | 0.0280 | 0.0380 | 0.0081 | 0.5350 | 0.8278 | 0.8427 |
| (14) | 1000 | 0.6112 | 0.2497 | 0.0537 | 0.6995 | 0.9140 | 0.9170 |
| | 5000 | 0.4800 | 0.0915 | 0.0130 | 0.5286 | 0.9129 | 0.9292 |
| | 10000 | 0.3638 | 0.0479 | 0.0078 | 0.5310 | 0.9231 | 0.9097 |

**Computational Efficiency:** Next we compare the computational efficiency of these competitors. We report the averages of the computing time in Table 2. In the present context we assume the observations arrive sequentially as $\{(\mathbf{x}_t, Y_t), t = 1, \ldots, \}$. For fair comparison, Chavent et al. (2014) suggested to implement the classic sliced inverse regression (M3) each time when a new observation arrives. It can be clearly seen that, the online learners are much faster than the batch learner (M3). In addition, the online learners (M2) and (M6) obtained through the gradient descent optimization are the fastest. This echoes our theoretical analysis in Proposition 4.

We further compare the computational efficiency of these competitors when the data streams are so massive that the batch learner hits the memory limit. In this case, the batch learner, which processes all observations simultaneously, cannot be used any more. The online learners, which update the estimates each time when a new instance arrives, have to be used instead. In other words, the online learners possess an additional advantage of low memory requirement. The averaged computing time of online learners are summarized in Table 3. These simulation results based on model (12). The second column of Table 3 gives the maximum sample sizes that the batch learner can process under the memory constraint

Table 2: The averaged computation time (in seconds) based on 100 replications. Refer to the caption of Table 1 for (M1)-(M6).

| Models | sample size | (M1) | (M2) | (M3) | (M4) | (M5) | (M6) |
|---|---|---|---|---|---|---|---|
| (12) | 1000 | 12.3209 | 0.3188 | 60.4872 | 0.4240 | 12.3371 | 0.2302 |
| | 5000 | 30.8054 | 0.8229 | 320.073 | 1.1961 | 31.3361 | 0.5918 |
| | 10000 | 59.2208 | 1.5360 | 1204.12 | 2.0451 | 60.0115 | 1.1051 |
| (13) | 1000 | 6.0793 | 0.1570 | 34.9824 | 0.2007 | 6.0796 | 0.1125 |
| | 5000 | 27.9957 | 0.7352 | 307.925 | 0.9260 | 28.4534 | 0.5253 |
| | 10000 | 52.9072 | 1.4286 | 1003.98 | 1.7310 | 55.0116 | 1.0429 |
| (14) | 1000 | 1.3369 | 0.1020 | 3.9825 | 0.1692 | 1.3191 | 0.0749 |
| | 5000 | 6.6453 | 0.5110 | 70.2830 | 0.8268 | 6.6539 | 0.3763 |
| | 10000 | 14.5415 | 1.1511 | 210.439 | 1.9740 | 14.6161 | 0.8499 |

given in the first column. The simulation results in the last five columns of Table 3 carry similar messages to those in Table 2, again indicating that the gradient descent optimization is more efficient than the perturbation method.

Table 3: The averaged computation time based on model (12). The second column gives the maximum sample sizes that the batch learner can process under the memory constraint given in the first column. Refer to the caption of Table 1 for (M1)-(M6).

| memory | sample sizes | (M1) | (M2) | (M4) | (M5) | (M6) |
|---|---|---|---|---|---|---|
| 1GB | $1 \times 10^7$ | 1.2 hour | 25.0 min | 24.1 min | 1.1 hour | 21.0 min |
| 2GB | $2 \times 10^7$ | 2.4 hour | 25.6 min | 51.8 min | 2.0 hour | 22.7 min |
| 4GB | $4 \times 10^7$ | 9.8 hour | 54.3 min | 1.7 hour | 8.6 hour | 46.7 min |
| 8GB | $9 \times 10^7$ | 14.4 hour | 1.8 hour | 3.7 hour | 13.2 hour | 1.5 hour |
| 16GB | $1.8 \times 10^8$ | 25.3 hour | 3.9 hour | 7.5 hour | 20.4 hour | 3.2 hour |

### 3.2 Real Data Analysis

In this section we further compare the performance of batch and online learners through real world applications. We consider 4 classification problems. To be specific, in the "wbc" and "magic04" data sets, the response is binary and the kernel matrices of sliced inverse regression have rank at most one. The "digits" data set aims to identify 10 hand-written digits $\{0, 1, \ldots, 9\}$, and the "ditigs069" data set aims to identify three similar digits $\{0, 6, 9\}$. The latter is a subset of the former. We further consider 4 regression problems. In particular, the housing data is available at http://lib.stat.cmu.edu/datasets/boston, the "abalone male" and "abalone female" data sets are available at http://archive.ics.uci.edu/ml, and the "ozone" data set is available at https://www.stat.umn.edu/arc/software.html. A brief description of these datasets are provided in Table 4. In these data sets, we estimate $K$

with the BIC type criterion. The estimated structural dimension $\widehat{K}$ is given in the last column of Table 4.

Table 4: The first two columns give, respectively, the name of each data set and the general task of usual data analysis. The third and the fourth columns stand for the sample sizes and the covariate dimensions, and the last column gives the estimated structural dimension $\widehat{K}$.

| data set | task | size | $p$ | $\widehat{K}$ |
|---|---|---|---|---|
| wbc | classification | 699 | 9 | 1 |
| magic04 | classification | 19020 | 10 | 1 |
| digits069 | classification | 2219 | 16 | 2 |
| digits | classification | 7494 | 16 | 4 |
| housing | Regression | 506 | 14 | 2 |
| abalone male | Regression | 1528 | 8 | 1 |
| abalone female | Regression | 1307 | 8 | 1 |
| ozone | Regression | 330 | 9 | 1 |

To compare the estimation accuracy of the online learners, we use the batch learner as a benchmark. We denote by $\widehat{\mathbf{B}}_0$ the estimate obtained with the batch learner, and by $\widehat{\mathbf{B}}$ the estimated obtained with the online learners. We calculate the distance $d(\widehat{\mathbf{B}}_0, \widehat{\mathbf{B}})$ between $\widehat{\mathbf{B}}_0$ and $\widehat{\mathbf{B}}$ using (15). We report the averaged distances of the online learners, (M1)-(M2) and (M4)-(M6), based on 100 permutations in Table 5. The online learners (M1) and (M2) appear closer to the batch learner (M3) than both the online learner (M4) and the unsupervised learners (M5)-(M6) in most situations.

Table 5: The averaged distances $d(\widehat{\mathbf{B}}_0, \widehat{\mathbf{B}})$ based on 100 permutations. The estimate $\widehat{\mathbf{B}}_0$ is obtained with the batch learner (M3), and the estimate $\widehat{\mathbf{B}}$ is obtained with the online learners. Refer to the caption of Table 1 for (M1)-(M6).

| data set | (M1) | (M2) | (M4) | (M5) | (M6) |
|---|---|---|---|---|---|
| wbc | 0.0371 | 0.1329 | 0.5500 | 0.3775 | 0.1113 |
| magic04 | 0.2969 | 0.4428 | 0.2444 | 0.9941 | 0.9955 |
| digits069 | 0.1377 | 0.0625 | 0.7232 | 0.5994 | 0.6054 |
| digits | 0.3190 | 0.0695 | 0.8320 | 0.8393 | 0.7687 |
| housing | 0.3848 | 0.2031 | 0.9803 | 0.9997 | 0.9999 |
| abalone male | 0.2585 | 0.1369 | 0.3241 | 0.8410 | 0.8404 |
| abalone female | 0.3309 | 0.2277 | 0.2999 | 0.8937 | 0.8973 |
| ozone | 0.4826 | 0.4215 | 0.8164 | 0.9778 | 0.9777 |

Next we compare the prediction accuracy of all online and batch learners (M1)-(M6). Towards this goal, we randomly select 75% of the observations as a training set and the remaining 25% as a test set. We use the SVM algorithm implemented in the R package e1071

(Meyer et al., 2017) to learn classifiers and build up regression models. We further build up a prediction model using all the original $p$ covariates without dimension reduction. We refer to it as (M7). To evaluate the prediction performance, we use the misclassification error rate in classification problems, which is defined as the proportion of incorrectly predicted labels in the test sets, namely,

$$\sum_{i\in\{\text{test}\}} \mathbb{1}(Y_i \neq \widehat{Y}_i) \Big/ \text{size of test set.}$$

In regression problems, we use the relative prediction error, which is defined as

$$\sum_{i\in\{\text{test}\}} (Y_i - \widehat{Y}_i)^2 \Big/ \sum_{i\in\{\text{test}\}} (Y_i - \overline{Y})^2.$$

The above procedure is repeated 100 times. We report the averages of prediction errors in Table 6. Our proposed online learners, (M1) and (M2), are comparable to the batch learning (M3) in classification problems, and significantly superior to the unsupervised learners (M5)-(M6) in regression problems. In addition, (M1) and (M2) performs better than (M7) which uses all $p$ covariates without dimension reduction in the last regression problems. This indicates that, if there exists a low dimensional structure, dimension reduction does enhances the prediction power. The unsupervised online learners, (M5) and (M6), exhibits relatively high predictive power in the "wbc" and "digits" data sets, partly because the covariates are highly correlated. Similar phenomenon is observed in Artemiou and Li (2009).

Table 6: The averaged prediction errors based on 100 repetitions. Refer to the caption of Table 1 for (M1)-(M6). (M7) stands for the prediction model built up with SVM using all the original $p$ covariates without dimension reduction.

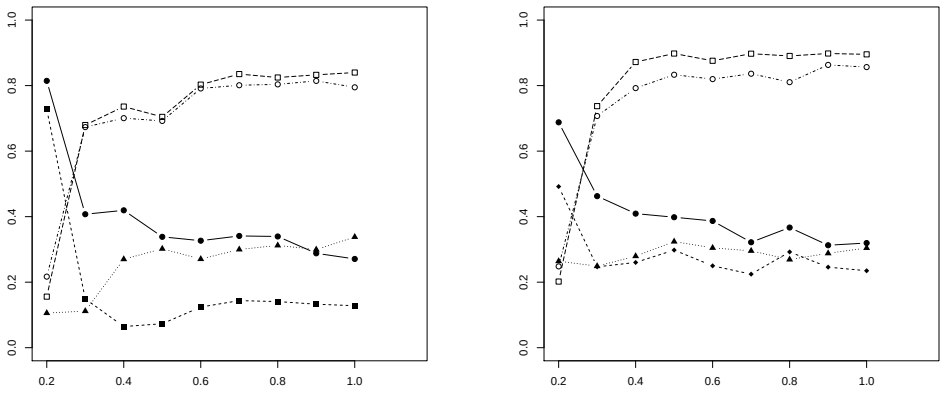
| data set | (M1) | (M2) | (M3) | (M4) | (M5) | (M6) | (M7) |
|---|---|---|---|---|---|---|---|
| wbc | 0.0351 | 0.0477 | 0.0330 | 0.0822 | 0.0770 | 0.0313 | 0.0357 |
| magic04 | 0.2111 | 0.2085 | 0.2073 | 0.2238 | 0.3170 | 0.3333 | 0.1329 |
| digits069 | 0.0115 | 0.0154 | 0.0135 | 0.0135 | 0.1188 | 0.0502 | 0.0096 |
| digits | 0.1126 | 0.1026 | 0.1063 | 0.1443 | 0.0623 | 0.0648 | 0.0208 |
| housing | 0.3353 | 0.3373 | 0.2338 | 0.4109 | 0.7725 | 0.7801 | 0.1867 |
| abalone male | 0.3624 | 0.3953 | 0.3115 | 0.1899 | 0.5966 | 0.5976 | 0.4389 |
| abalone female | 0.3445 | 0.3729 | 0.2841 | 0.2024 | 0.5985 | 0.5972 | 0.4450 |
| ozone | 0.5725 | 0.4844 | 0.6556 | 0.2077 | 0.5201 | 0.5209 | 0.7468 |

In our previous analysis, we use the SVM to build up a prediction model. In what follows, we compare SVM with the classification tree, linear discriminant analysis, generalized linear model, and random forest, to build up prediction models. We use the " digits" data as an example. The prediction errors are summarized in Table 7. All these numbers exhibit very similar patterns, although the random forest and the support vector machine exhibit the highest prediction power. This indicates that using the SVM to build a prediction model does not have an essential effect in our comparative studies.

Table 7: The prediction errors based on the "digits" data. The models are built up with support vector machines (SVM), classification tree (TREE), linear discriminant analysis (LDA), generalized linear model (GLM), and random forest (RF).

| method | (M1) | (M2) | (M3) | (M4) | (M5) | (M6) | (M7) |
|---|---|---|---|---|---|---|---|
| SVM | 0.1126 | 0.1026 | 0.1063 | 0.1443 | 0.0623 | 0.0648 | 0.0208 |
| TREE | 0.3530 | 0.2372 | 0.2830 | 0.4505 | 0.2710 | 0.2458 | 0.2612 |
| LDA | 0.2415 | 0.2035 | 0.2026 | 0.3130 | 0.2747 | 0.2475 | 0.1703 |
| GLM | 0.2038 | 0.1589 | 0.1532 | 0.2767 | 0.2449 | 0.1961 | 0.1037 |
| RF | 0.1069 | 0.0963 | 0.1043 | 0.1535 | 0.0663 | 0.0706 | 0.0345 |

We further investigate the performance of the online learners with data streams are evolving. We use (15) to evaluate the distance between the estimates obtained with the batch and online learners. We use the "abalone male" and "abalone female" data sets as examples. It can be clearly seen from Figure 1 that, as the sample size is increased to a reasonable scale, say, 50% of the dataset, our proposed online learners, (M1) and (M2), already perform very steadily, and appear very close to the batch learner (M3).




(A): abalone male (B): abalone female

Figure 1: The horizontal axis denotes the proportion of sample size used by the batch and online learners, and the vertical axis denotes the distance between the estimates obtained with the batch and online learners. The solid lines marked with circles and squares represent online sliced inverse regression with perturbation method (M1) and gradient descent optimization (M2), respectively. The solid line marked with triangles represents sliced inverse regression via block-wise learning (M4). The hollow line marked with circles and squares represent online principal component analysis via perturbation method (M5) and gradient descent optimization (M6), respectively.

## 4. Concluding Remarks

In this paper we propose to implement sliced inverse regression in an online fashion. This procedure consists of two steps. In the first step we construct an online estimate for the kernel matrix of sliced inverse regression. Towards this goal, we modify sliced inverse regression slightly to reduce the computational complexity of the online learners. This modification indeed leads a variation of cumulative slicing estimation. In the second step we propose two algorithms, one is motivated by the perturbation method and the other is originated from the gradient descent optimization, to perform online singular value decomposition. We investigate thoroughly the theoretical convergence properties of the online learners.

## Acknowledgments

The authors would like to thank the action editor and reviewers for their constructive comments, which lead to a significant improvement of this work. All authors made equal contribution to this work, the authors are listed in the alphabetic order, and Runze Li is the corresponding author. Cai and Li's research was supported by an NSF grant DMS 1820702. Zhu's research was supported by Beijing Natural Science Foundation (Z190002) and National Natural Science Foundation of China (11731011, 11931014, 11690015).

## Appendix A. Technical Lemmas

To prove Theorem 2, we state a lemma on quasi-martingales, which provide sufficient conditions of convergence for a stochastic process.

**Lemma 5** *(Bottou, 1998; Fisk, 1965) Define $(\Omega, \mathcal{F}, P)$ to be a measurable probability space. Let $w_t$, $t > 0$, be the realization of a stochastic process and $\mathcal{F}_t$ be the filtration determined by the past information at time $t$. Let $\varepsilon_t = \mathbb{1}\{E(w_{t+1} - w_t \mid \mathcal{F}_t) > 0\}$ If for all $t$, $w_t \geq 0$ and $\sum_{t=1}^{\infty} E\{\varepsilon_t(w_{t+1} - w_t)\} < \infty$, then $w_t$ is a quasi-martingale and converges almost surely. Moreover, $\sum_{t=1}^{\infty} |E(w_{t+1} - w_t \mid \mathcal{F}_t)| < +\infty$ almost surely.*

The following three lemmas are essential to the proof of Theorem 3. The ideas are originated from Oja and Karhunen (1985). Let $(\lambda_j, \boldsymbol{\beta}_j)$, $j = 1, \cdots, K$ be the corresponding eigenpairs of $\mathbf{M}$. Firstly, we consider the case of $K = 1$, where $\widehat{\mathbf{B}}_t$ only consists of $\widehat{\boldsymbol{\beta}}_{t,1}$. Recall that for the time $t$ update, the gradient descent algorithm can be written as $\widehat{\boldsymbol{\beta}}_{t+1,1} = \mathcal{P}_{\text{orth}}(\widehat{\boldsymbol{\beta}}_{t,1} + \gamma_{t+1}\widehat{\mathbf{M}}_t\widehat{\boldsymbol{\beta}}_{t,1})$. We write $\widehat{\boldsymbol{\beta}}_{t,1}$ as $\widehat{\boldsymbol{\beta}}_t$ with slightly abuse of notations. By using Taylor expansion, this equation can be expressed as a power series in $\gamma_{t+1}$

$$\widehat{\boldsymbol{\beta}}_{t+1} = \widehat{\boldsymbol{\beta}}_t + \gamma_{t+1}\big\{\widehat{\mathbf{M}}_t\widehat{\boldsymbol{\beta}}_t - (\widehat{\boldsymbol{\beta}}_t^\top \mathbf{M}_t\widehat{\boldsymbol{\beta}}_t)\widehat{\boldsymbol{\beta}}_t\big\} + \gamma_{t+1}\mathbf{b}_t$$

where $\mathbf{b}_t = o(\gamma_{t+1})$. Since $\widehat{\boldsymbol{\beta}}_t^\top\widehat{\boldsymbol{\beta}}_t = 1$, this equation can be further rewritten as

$$\widehat{\boldsymbol{\beta}}_{t+1} = \widehat{\boldsymbol{\beta}}_t + \gamma_{t+1}\left(\widehat{\mathbf{M}}_t\widehat{\boldsymbol{\beta}}_t - \frac{\widehat{\boldsymbol{\beta}}_t^\top\widehat{\mathbf{M}}_t\widehat{\boldsymbol{\beta}}_t}{\widehat{\boldsymbol{\beta}}_t^\top\widehat{\boldsymbol{\beta}}_t}\widehat{\boldsymbol{\beta}}_t\right) + \gamma_{t+1}\mathbf{b}_t$$

We construct the next Lemma to deal with the dependence structure of $\widehat{\mathbf{M}}_t$. It shows that even without the independence assumption, the tail of sum of root-$n$ convergent sequences converges to zero.

**Lemma 6** *Assume that a sequence of non-negative random variables* $\{r_n\}$ *satisfy* $r_n = O_p(n^{-1/2})$. *Then* $\forall \varepsilon > 0$

$$\lim_{k\to\infty} \Pr\left(\sum_{n=k}^{\infty} n^{-1} r_n > \varepsilon\right) = 0$$

*Proof* $\forall \varepsilon > 0$, write $\varepsilon = \varepsilon(\sum_{n=k}^{\infty} n^{-3/2})H$, where $H = (\sum_{n=k}^{\infty} n^{-3/2})^{-1}$. Since $\sum_{n=k}^{\infty} n^{-3/2} \sim O(k^{-1/2})$, it is true that $H \sim O(k^{1/2})$. On the other hand, we have

$$\begin{aligned}\Pr(\sum_{n=k}^{\infty} n^{-1} r_n > \varepsilon) &= \Pr(\sum_{n=k}^{\infty} n^{-1} r_n > \sum_{n=k}^{\infty} \varepsilon n^{-3/2} H) \\ &\le \Pr(n^{-1} r_n > \varepsilon n^{-3/2} H) = \Pr(n^{1/2} r_n > \varepsilon H)\end{aligned}$$

for some $n > k$. And by the definition for $r_n = O_p(n^{-1/2})$, it is true that for $\forall \widetilde{\varepsilon} > 0$, there exists constants $M$ and $N$, such that for $\forall n > N$

$$\Pr(n^{1/2} r_n > M) < \widetilde{\varepsilon}$$

Since $\varepsilon > 0$ and $H \sim O(k^{1/2})$, we can take $k$ large enough such that $k > N$ and $\varepsilon H > M$. Combine those two inequalities above, we have $\Pr(\sum_{n=k}^{\infty} n^{-1} r_n > \varepsilon) < \widetilde{\varepsilon}$, $\forall \widetilde{\varepsilon} > 0$. ■

The following Lemma 7 is a slightly modified result of Theorem 2.3.1 from Kushner and Clark (2012), which is also similar with Lemma 1 from Oja and Karhunen (1985). Thus we only need to verify the technical conditions.

**Lemma 7** *Assume conditions C2-C4 hold. Let* $\mathbf{z}_0$ *be a locally asymptotically stable (in the sense of Liapunov) solution to*

$$\frac{d\mathbf{z}}{dt} = \mathbf{M}\mathbf{z} - \frac{(\mathbf{z}^\top \mathbf{M}\mathbf{z})\mathbf{z}}{\mathbf{z}^\top \mathbf{z}} \tag{16}$$

*with domain of attraction* $\mathcal{D}(\mathbf{z}_0)$. *If there is a compact set* $\mathscr{A} \subset \mathcal{D}(\mathbf{z}_0)$ *such that the solution* $\widehat{\boldsymbol{\beta}}_t \in \mathscr{A}$ *infinitely often, then* $\widehat{\boldsymbol{\beta}}_t$ *tends to* $\mathbf{z}_0$ *almost surely.*

*Proof* We verify the conditions of Theorem 2.3.1 from Kushner and Clark (2012) as follows. Assumptions A.2.2.1 and A.2.2.3 are due to (16) and condition C4. The boundedness of $\widehat{\boldsymbol{\beta}}_t$ is due to the projection onto the orthonormal space. The reminder term $\mathbf{b}_t$ can further be expanded as

$$\mathbf{b}_t = -\frac{1}{2}\gamma_{t+1}(\widehat{\boldsymbol{\beta}}_t^\top \widehat{\mathbf{M}}_t^2 \widehat{\boldsymbol{\beta}}_t)\widehat{\boldsymbol{\beta}}_t - \frac{1}{2}\gamma_{t+1}\alpha_t \widehat{\mathbf{M}}_t \widehat{\boldsymbol{\beta}}_t + \gamma_{t+1}^{-1}\{(1+\gamma_{t+1}\alpha_t)^{-1/2} - 1 + \frac{1}{2}\gamma_{t+1}\alpha_t\}(I - \gamma_{t+1}\widehat{\mathbf{M}}_t)\widehat{\boldsymbol{\beta}}_t$$

where $\alpha_t = 2\widehat{\boldsymbol{\beta}}_t^\top \widehat{\mathbf{M}}_t \widehat{\boldsymbol{\beta}}_t + \gamma_{t+1}\widehat{\boldsymbol{\beta}}_t^\top \widehat{\mathbf{M}}_t^2 \widehat{\boldsymbol{\beta}}_t$. $\mathbf{b}_t$ is a.s. bounded and tends to zero as $\gamma_{t+1} \to 0$ because both $\widehat{\boldsymbol{\beta}}_t$ and $\widehat{\mathbf{M}}_t$ are a.s. bounded. Thus condition A.2.2.2 is verified. As for

condition A.2.2.4, we first control the following term into sum of univariate variables as follows:

$$\begin{aligned}
&\sup_{m\geq k} \| \sum_{i=k}^{m} \gamma_i \{ (\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i - \widehat{\boldsymbol{\beta}}_i^\top (\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i \} \|_1 \\
\leq &\sup_{m\geq k} \sum_{i=k}^{m} \gamma_i \{ \mathbf{1}_p^\top |(\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i| + \mathbf{1}_p^\top |\widehat{\boldsymbol{\beta}}_i^\top (\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i| \} \\
\leq &\sum_{i=k}^{\infty} \gamma_i \{ \mathbf{1}_p^\top |(\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i| + \mathbf{1}_p^\top |\widehat{\boldsymbol{\beta}}_i^\top (\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i| \}
\end{aligned}$$

Let $r_t = \mathbf{1}_p^\top |(\widehat{\mathbf{M}}_t - \mathbf{M})\widehat{\boldsymbol{\beta}}_t| + \mathbf{1}_p^\top |\widehat{\boldsymbol{\beta}}_t^\top (\widehat{\mathbf{M}}_t - \mathbf{M})\widehat{\boldsymbol{\beta}}_t \widehat{\boldsymbol{\beta}}_t|$, we have $r_t = O_p(t^{-1/2})$ as a result of the root-$t$ consistency of $\widehat{\mathbf{M}}_t$. Thus

$$\Pr(\sup_{m\geq k} \| \sum_{i=k}^{m} \gamma_i \{ (\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i - \widehat{\boldsymbol{\beta}}_i^\top (\widehat{\mathbf{M}}_i - \mathbf{M})\widehat{\boldsymbol{\beta}}_i \widehat{\boldsymbol{\beta}}_i \} \|_1 \geq \varepsilon) \leq \Pr(\sum_{n=k}^{\infty} \gamma_i r_n \geq \varepsilon)$$

In the meanwhile, Lemma 6 implies $\Pr(\sum_{n=k}^{\infty} \gamma_i r_n \geq \varepsilon) \to 0$ as $k \to \infty$. Thus condition A.2.2.4 holds true. The proof is completed by Theorem 2.3.1 of Kushner and Clark (2012).■

**Lemma 8** *(Oja and Karhunen, 1985) Assume C2 holds. The points $\boldsymbol{\beta}_1$ and $-\boldsymbol{\beta}_1$ are uniformly asymptotically stable. The domain of attraction of $\boldsymbol{\beta}_1$ is $\mathcal{D}(\boldsymbol{\beta}_1) = \{\mathbf{x} \in \mathbb{R}^p | \mathbf{x}^\top \boldsymbol{\beta}_1 > 0\}$ while for $-\boldsymbol{\beta}_1$ is $\mathcal{D}(-\boldsymbol{\beta}_1) = \{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{x}^\top \boldsymbol{\beta}_1 < 0\}$.*

## Appendix B: Proof of Theorem 2

Firstly, we show the almost surely convergence of $(\widehat{\lambda}_{t,1}, \ldots, \widehat{\lambda}_{t,K})$. For the $j$-th element $\widehat{\lambda}_{t,j}$, the update formula is given by (9). Denote $d_{jt}$ as $\widehat{\boldsymbol{\beta}}_{t,j}^\top (\widehat{\boldsymbol{\Gamma}}_t - \widehat{\mathbf{M}}_{t+1})\widehat{\boldsymbol{\beta}}_{t,j}$ with slightly abuse of notation. Thus the update of eigenvalues can be rewritten as

$$\widehat{\lambda}_{t+1,j} = \widehat{\lambda}_{t,j} - (t+1)^{-1} d_{jt}$$

To apply the lemma on quasi-martingales, we first show that $t\mathbb{E}[|d_{jt}|]$ is uniformly bounded. We begin with the expression of $|d_{jt}|$,

$$\begin{aligned}
|d_{jt}| &= |\widehat{\boldsymbol{\beta}}_{t,j}^\top (\frac{1}{t}\sum_{i=1}^{t} \widehat{\mathbf{M}}_i - \widehat{\mathbf{M}}_{t+1})\widehat{\boldsymbol{\beta}}_{t,j}| \\
&\leq |\widehat{\boldsymbol{\beta}}_{t,j}^\top| |(\frac{1}{t}\sum_{i=1}^{t} (\widehat{\mathbf{M}}_i - \mathbf{M}) - (\widehat{\mathbf{M}}_{t+1} - \mathbf{M}))| |\widehat{\boldsymbol{\beta}}_{t,j}| \\
&\leq \mathbf{m}^\top |(\frac{1}{t}\sum_{i=1}^{t} (\widehat{\mathbf{M}}_i - \mathbf{M}) - (\widehat{\mathbf{M}}_{t+1} - \mathbf{M}))| \mathbf{m} \\
&\leq \mathbf{m}^\top |\frac{1}{t}\sum_{i=1}^{t} (\widehat{\mathbf{M}}_i - \mathbf{M})| \mathbf{m} + \mathbf{m}^\top |(\widehat{\mathbf{M}}_{t+1} - \mathbf{M})| \mathbf{m}
\end{aligned}$$

where we used the fact that the absolute value of the orthogonal eigenvectors $\widehat{\boldsymbol{\beta}}_{t,j}$ can be uniformly bounded by a positive vector $\mathbf{m}$. Since $\widehat{\mathbf{M}}_t - \mathbf{M} = \sum_{h=1}^{H}(\widehat{\mathbf{m}}_{t,h}\widehat{\mathbf{m}}_{t,h}^\top - \mathbf{m}_h\mathbf{m}_h^\top)$ and $H$ is a finite constant, it suffices to study $\mathbb{E}(\widehat{\mathbf{m}}_{t,h}\widehat{\mathbf{m}}_{t,h}^\top - \mathbf{m}_h\mathbf{m}_h^\top)$. Define $\boldsymbol{\varepsilon}_h = \mathbb{1}(Y_t \in I_h) - \mathbf{1}_{t\times 1}\alpha - \mathbf{X}\mathbf{m}_h$. By condition C2 we have $\mathbb{E}\boldsymbol{\varepsilon}_h = \mathbb{0}$ and $\mathrm{Var}(\boldsymbol{\varepsilon}_h) = \sigma^2\mathbf{I}$, where $\sigma^2$ is a constant and $\mathbf{I}$ is the identity matrix. Let $\widetilde{\mathbf{X}} = (\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_t)^\top \in \mathbb{R}^{t\times(p+1)}$. By the formulation and the online algorithm for the kernel matrix estimate in Section 2, we know that $\widehat{\mathbf{m}}_{t,h} = \widetilde{\mathbf{I}}_{p\times(p+1)}(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^\top\mathbb{1}(Y_t \in I_h)$. Thus $\widehat{\mathbf{m}}_{t,h}\widehat{\mathbf{m}}_{t,h}^\top = \widetilde{\mathbf{I}}_{p\times(p+1)}(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^\top(\widetilde{\mathbf{X}}\mathbf{m}_h+\boldsymbol{\varepsilon}_h)(\widetilde{\mathbf{X}}\mathbf{m}_h+\boldsymbol{\varepsilon}_h)^\top\top\widetilde{\mathbf{X}}^\top(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{I}}_{p\times(p+1)}^\top$. It follows that

$$\begin{aligned}\widehat{\mathbf{m}}_{t,h}\widehat{\mathbf{m}}_{t,h}^\top - \mathbf{m}_h\mathbf{m}_h^\top \propto&(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^\top\boldsymbol{\varepsilon}_h\boldsymbol{\varepsilon}_h^\top\widetilde{\mathbf{X}}(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}+\\ &\mathbf{m}_h\boldsymbol{\varepsilon}_h^\top\widetilde{\mathbf{X}}(\widetilde{\mathbf{X}}^\top\mathbf{X})^{-1} + (\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^\top\boldsymbol{\varepsilon}_h\mathbf{m}_h^\top\\ \triangleq& S_1 + S_2 + S_2^\top\end{aligned}$$

Notice that $\mathbb{E}[S_1|\widetilde{\mathbf{X}}] = \sigma^2(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}$, which is of order $t^{-1}$ by condition C1. $\mathbb{E}[S_2|\widetilde{\mathbf{X}}] = \mathbf{0}$. Thus $\mathbb{E}[\widehat{\mathbf{m}}_{t,h}\widehat{\mathbf{m}}_{t,h}^\top - \mathbf{m}_h\mathbf{m}_h^\top] = \mathbb{E}[\mathbb{E}\{(\widehat{\mathbf{m}}_{t,h}\widehat{\mathbf{m}}_{t,h}^\top - \mathbf{m}_h\mathbf{m}_h^\top)|\widetilde{\mathbf{X}}\}]$ is also of order $t^{-1}$. This shows that $\mathbb{E}[|d_{jt}|]$ can be uniformly bounded by $C/t$, where $C$ is a constant. Therefore, define $\varepsilon_t$ as in Lemma 5, it follows that

$$\sum_{t=1}^{\infty}\mathbb{E}[\varepsilon_t(\widehat{\lambda}_{t+1,j} - \widehat{\lambda}_{t,j})] \le \sum_{t=1}^{\infty}\mathbb{E}[\frac{1}{t+1}|d_{jt}|] \le \sum_{i=1}^{t}\frac{C}{t(t+1)} < \infty$$

It follows from Lemma 5 that $\widehat{\lambda}_{t,j}$ is a quasi-martingale and converges almost surely, which holds true for $j = 1, \ldots, K$. The proof of strong convergence for the eigenvectors $\widehat{\boldsymbol{\beta}}_{t,j}$ proceed along with very similar lines and is omitted here. This completes the proof. ■

## Appendix C: Proof of Theorem 3

Consider the case of $K = 1$. Firstly, we show that there exists a number $\varepsilon$ such that the event $|\widehat{\boldsymbol{\beta}}_t^\top\boldsymbol{\beta}_1| > \varepsilon$ occurs infinitely often almost surely. From (11), we have

$$\widehat{\boldsymbol{\beta}}_{t+1}^\top\boldsymbol{\beta}_1 = \frac{\widehat{\boldsymbol{\beta}}_t^\top\boldsymbol{\beta}_1 + \gamma_{t+1}\boldsymbol{\beta}_1^\top\{(\widehat{\mathbf{M}}_t - \mathbf{M}) + \mathbf{M}\}\widehat{\boldsymbol{\beta}}_t}{(1+\gamma_{t+1}\widehat{\mathbf{M}}_t)\widehat{\boldsymbol{\beta}}_t}$$

Without loss of generality, we assume that $\widehat{\boldsymbol{\beta}}_t^\top\boldsymbol{\beta}_1 > 0$. Because $\widehat{\mathbf{M}}_t$ is an asymptotically consistent estimate of $\mathbf{M}$, there exist positive numbers $\delta$ and $\rho$ such that for $t$ large enough, $\Pr\{\boldsymbol{\beta}_1^\top(\widehat{\mathbf{M}}_t - \mathbf{M})\widehat{\boldsymbol{\beta}}_t \ge \delta\} \ge \rho$ uniformly. Denote $\eta$ to be the almost sure upper bound for $\widehat{\mathbf{M}}_t$ and also let $\eta$ be larger than $\lambda_1$. Thus

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{t+1}^\top\boldsymbol{\beta}_1 \ge&(1+\gamma_{t+1}\eta)^{-1}(\widehat{\boldsymbol{\beta}}_t^\top\boldsymbol{\beta}_1 + \gamma_{t+1}\delta + \gamma_{t+1}\lambda_1\boldsymbol{\beta}_1^\top\widehat{\boldsymbol{\beta}}_t)\\ =&\frac{1+\gamma_{t+1}\lambda_1}{1+\gamma_{t+1}\eta}\boldsymbol{\beta}_1^\top\widehat{\boldsymbol{\beta}}_t + \frac{\gamma_{t+1}}{1+\gamma_{t+1}\eta}\delta\end{aligned}$$

Because $\widehat{\mathbf{M}}_t$ is highly correlated, there exists a positive number $\rho_m$ such that with probability at least $\rho_m$, $\boldsymbol{\beta}_1^\top(\widehat{\mathbf{M}}_n - \mathbf{M})\widehat{\boldsymbol{\beta}}_n \ge \delta$ for all $n = t, t+1, \ldots, t+m$. It follows that

$$\widehat{\boldsymbol{\beta}}_{t+1}^\top\boldsymbol{\beta}_1 \ge \frac{1+\gamma_{n+1}\lambda_1}{1+\gamma_{n+1}\eta}\boldsymbol{\beta}_1^\top\widehat{\boldsymbol{\beta}}_n + \frac{\gamma_{n+1}}{1+\gamma_{n+1}\eta}\delta, \quad n = t, \ldots, t+m$$

which gives

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{t+m}^{\top}\boldsymbol{\beta}_1 \geq & \prod_{n=t}^{t+m}\left(\frac{1+\gamma_{n+1}\lambda_1}{1+\gamma_{n+1}\eta}\right)\boldsymbol{\beta}_1^{\top}\widehat{\boldsymbol{\beta}}_n + \delta\sum_{n=t}^{t+m}\left(\frac{\gamma_{n+1}}{1+\gamma_{n+1}\eta}\right)\prod_{i=n+1}^{t+m}\left(\frac{1+\gamma_i\lambda_1}{1+\gamma_i\eta}\right)\\ \geq & \delta\sum_{n=t}^{t+m}\left(\frac{\gamma_{n+1}}{1+\gamma_{n+1}\eta}\right)\prod_{i=n+1}^{t+m}\left(\frac{1+\gamma_i\lambda_1}{1+\gamma_i\eta}\right)\end{aligned}$$

Since $\gamma_t = t^{-1}$, we may assume without loss of generality that $0 \leq \gamma_{n+1} \leq \lambda_1^{-1}$ for all $n \geq t$. Define the product of the form $\prod_{i=t+m+1}^{t+m}$ to have the value 1, we have

$$\begin{aligned}&\delta\sum_{n=t}^{t+m}\left(\frac{\gamma_{n+1}}{1+\gamma_{n+1}\eta}\right)\prod_{i=n+1}^{M}\left(\frac{1+\gamma_i\lambda_1}{1+\gamma_i\eta}\right)\\ =&\frac{\delta}{\eta-\lambda_1}\sum_{n=t}^{t+m}\left(\frac{(1+\gamma_{n+1}\eta)-(1+\gamma_{n+1}\lambda_1)}{1+\gamma_{n+1}\eta}\right)\prod_{i=n+1}^{t+m}\left(\frac{1+\gamma_i\lambda_1}{1+\gamma_i\eta}\right)\\ =&\frac{\delta}{\eta-\lambda_1}\left\{1-\prod_{i=t}^{t+m}\left(\frac{1+\gamma_i\lambda_1}{1+\gamma_i\eta}\right)\right\}\end{aligned}$$

The fact that $\eta > \lambda_1$ ensures there exists a positive constant $\theta$ such that $e^{-\theta w} \geq (1 + w\lambda_1)/(1 + w\alpha)$ for $w \in [0, \lambda^{-1}]$. Therefore, $(1 + \gamma_i\lambda_1)/(1 + \gamma_i\alpha) \leq e^{-\theta\gamma_i}$ for $i = t, t + 1 \ldots, t + m$. This gives

$\prod_{i=t}^{t+m}\left(\frac{1+\gamma_i\lambda_1}{1+\gamma_i\eta}\right) \leq \exp^{-\theta\sum_{i=t}^{t+m}\gamma_i}$. Consequently, $\widehat{\boldsymbol{\beta}}_{t+m}^{\top}\boldsymbol{\beta}_1 \geq \frac{\delta}{\eta-\lambda_1}\left(1-\exp^{-\theta\sum_{i=t}^{t+m}\gamma_i}\right)$

We choose $\varepsilon = \delta/2(\eta - \lambda_1)$.Since $\sum\gamma_i$ is divergent, we can always find a $m$ such that

$$\frac{\delta}{\eta-\lambda_1}\left(1-\exp^{-\theta\sum_{i=t}^{t+m}\gamma_i}\right) \geq \varepsilon.$$

Thus, the event $\widehat{\boldsymbol{\beta}}_{t+m}^{\top}\boldsymbol{\beta}_1 \geq \varepsilon$ happens with a positive probability $\rho_m$. Next we show that $\widehat{\boldsymbol{\beta}}_t$ is a Markov process. This is because by (11), we have

$$\widehat{\boldsymbol{\beta}}_{t+1} = \frac{\widehat{\boldsymbol{\beta}}_t + \gamma_{t+1}\widehat{\mathbf{M}}_t\widehat{\boldsymbol{\beta}}_t}{\|\widehat{\boldsymbol{\beta}}_t + \gamma_{t+1}\widehat{\mathbf{M}}_t\widehat{\boldsymbol{\beta}}_t\|}$$

where $\widehat{\mathbf{M}}_t$ plays an essential role. From $\widehat{\mathbf{M}}_t$'s update formula (5), we can verify that $\widehat{\mathbf{M}}_t$ is a Markov process because $\{(\mathbf{x}_t, Y_t), t = 1, \ldots\}$ are independent and identically distributed. Thus, $\widehat{\boldsymbol{\beta}}_t$ is also a Markov process. Starting from any state such that $\widehat{\boldsymbol{\beta}}_t^{\top}\boldsymbol{\beta}_1 > 0$, the region $\{\widetilde{\boldsymbol{\beta}} : \widetilde{\boldsymbol{\beta}}^{\top}\boldsymbol{\beta}_1 > 0\}$ will be eventually reached with probability one. Similarly, starting from $\widehat{\boldsymbol{\beta}}_t^{\top}\boldsymbol{\beta}_1 < 0$, the region $\{\widetilde{\boldsymbol{\beta}} : \widetilde{\boldsymbol{\beta}}^{\top}\boldsymbol{\beta}_1 < 0\}$ will also be reached with probability one. Thus, $\widehat{\boldsymbol{\beta}}_t$ visits infinitely often a compact subset of the domain of attraction, which is one of the asymptotically stable points $\boldsymbol{\beta}$ and $-\boldsymbol{\beta}$ in (16). By Lemma 7, $\widehat{\boldsymbol{\beta}}_t$ converges almost surely to $\boldsymbol{\beta}$ or $-\boldsymbol{\beta}$. For the case of $K > 1$, we refer the readers to Theorem 2 of Oja and Karhunen (1985). The proofs will follow in a similar way and here we omit them for simplicity. ■

## References

Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for PCA and PLS. In *50th Allerton Conference on Communication, Control, and Computing*, pages 861–868. IEEE, 2012.

Raman Arora, Andy Cotter, and Nati Srebro. Stochastic optimization of PCA with capped MSG. In *Advances in Neural Information Processing Systems*, pages 1815–1823, 2013.

Andreas Artemiou and Bing Li. On principal components and regression: a statistical explanation of a natural phenomenon. *Statistica Sinica*, pages 1557–1565, 2009.

Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems*, pages 3174–3182, 2013.

Bernard Bercu, Thi Mong Ngoc Nguyen, and Jerome Saracco. On the asymptotic behaviour of the recursive nadaraya–watson estimator associated with the recursive sliced inverse regression method. *Statistics*, 49(3):660–679, 2015.

Léon Bottou. *Online Learning and Stochastic Approximations*. Cambridge University Press, Cambridge, UK, 1998.

Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

Christos Boutsidis, Dan Garber, Zohar Karnin, and Edo Liberty. Online principal components analysis. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 887–901. SIAM, 2015.

Benoit Champagne. Adaptive eigendecomposition of data covariance matrices based on first-order perturbations. *IEEE Transactions on Signal Processing*, 42(10):2758–2770, 1994.

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.

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

R Dennis Cook and Sanford Weisberg. Discussion of sliced inverse regression for dimension reduction by Li,K.C. *Journal of the American Statistical Association*, 86(414):328–332, 1991.

Donald L Fisk. Quasi-martingales. *Transactions of the American Mathematical Society*, 120(3):369–389, 1965.

Anant Hegde, Jose C Principe, Deniz Erdogmus, Umut Ozertem, Yadunandana N Rao, and Hemanth Peddaneni. Perturbation-based eigenvector updates for on-line principal components analysis and canonical correlation analysis. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 45(1-2):85–95, 2006.

Tosio Kato. *Perturbation Theory for Linear Operators*, volume 132. Springer Science & Business Media, 2013.

Harold Joseph Kushner and Dean S Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, volume 26. Springer Science & Business Media, 2012.

Yunwen Lei, Lei Shi, and Zheng-Chu Guo. Convergence of unregularized online learning algorithms. *The Journal of Machine Learning Research*, 18(1):6269–6301, 2017.

Bing Li. *Sufficient Dimension Reduction: Methods and Applications with R*. CRC Press, 2018.

Bing Li and Shaoli Wang. On directional regression for dimension reduction. *Journal of the American Statistical Association*, 102(479):997–1008, 2007.

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

Weihua Li, H Henry Yue, Sergio Valle-Cervantes, and S Joe Qin. Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10(5):471–486, 2000.

Yanyuan Ma and Liping Zhu. A semiparametric approach to dimension reduction. *Journal of the American Statistical Association*, 107(497):168–179, 2012.

Yanyuan Ma and Liping Zhu. A review on dimension reduction. *International Statistical Review*, 81(1):134–150, 2013.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. e1071: misc functions of the department of statistics, probability theory group (formerly: E1071), TU Wien. R package version 1.6-8. 2017.

Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming PCA. In *Advances in Neural Information Processing Systems*, pages 2886–2894, 2013.

Jiazhong Nie, Wojciech Kotlowski, and Manfred K. Warmuth. Online PCA with optimal regret. *Journal of Machine Learning Research*, 17(173):1–49, 2016.

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.

Ohad Shamir. Convergence of stochastic gradient descent for PCA. In *International Conference on Machine Learning*, pages 257–265, 2016.

Robin Sibson. Studies in the robustness of multidimensional scaling: Perturbational analysis of classical scaling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 217–229, 1979.

Pierre Tarres and Yuan Yao. Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence. *IEEE Transactions on Information Theory*, 60(9):5716–5735, 2014.

Manfred K Warmuth and Dima Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9 (Oct):2287–2320, 2008.

Yingcun Xia, Howell Tong, Wai Keung 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.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Wenzhuo Yang and Huan Xu. Streaming sparse principal component analysis. In *International Conference on Machine Learning*, pages 494–503, 2015.

Zhishen Ye and Robert E Weiss. Using the bootstrap to select one of a new class of dimension reduction methods. *Journal of the American Statistical Association*, 98(464): 968–979, 2003.

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.

Li-Xing Zhu and Kai Wang Ng. Asymptotics of sliced inverse regression. *Statistica Sinica*, pages 727–736, 1995.