JMLR: Workshop and Conference Proceedings 30:1–16, 2015 ACML 2015

Sufficient Dimension Reduction via Direct Estimation of the Gradients of Logarithmic Conditional Densities

Hiroaki Sasaki SASAKI@MS.K.U-TOKYO.AC.JP
Graduate School of Frontier Sciences
The University of Tokyo
Chiba, Japan

Voot Tangkaratt VOOT@MS.K.U-TOKYO.AC.JP
Graduate School of Information Science and Technology
The University of Tokyo
Tokyo, Japan

Masashi Sugiyama SUGI@K.U-TOKYO.AC.JP
Graduate School of Frontier Sciences
The University of Tokyo
Chiba, Japan

Abstract

Sufficient dimension reduction (SDR) is a framework of supervised linear dimension reduction, and is aimed at finding a low-dimensional orthogonal projection matrix for input data such that the projected input data retains maximal information on output data. A computationally efficient approach employs gradient estimates of the conditional density of the output given input data to find an appropriate projection matrix. However, since the gradients of the conditional densities are typically estimated by a local linear smoother, it does not perform well when the input dimensionality is high. In this paper, we propose a novel estimator of the gradients of logarithmic conditional densities called the *least-squares logarithmic conditional density gradients* (LSLCG), which fits a gradient model *directly* to the true gradient without conditional density estimation under the squared loss. Thanks to the simple least-squares formulation, LSLCG gives a closed-form solution that can be computed efficiently. In addition, all the parameters can be automatically determined by cross-validation. Through experiments on a large variety of artificial and benchmark datasets, we demonstrate that the SDR method based on LSLCG outperforms existing SDR methods both in estimation accuracy and computational efficiency.

Keywords: sufficient dimension reduction, logarithmic conditional density, gradient estimation.

1. Introduction

High-dimensional data is ubiquitous in modern statistical data analysis, and overcoming the *curse of dimensionality* is a big challenge. To cope with the difficulty of handling high-dimensional data, *dimension reduction* is a common approach (Carreira-Perpiñán, 1997; Jain et al., 2000). For supervised dimension reduction, *sufficient dimension reduction* (SDR) (Li, 1991; Cook, 1998; Chiaromonte and Cook, 2002) is a well-established framework, which is aimed at finding a low-dimensional projection matrix for input data vectors so that the projected data contains as much information as possible on the output data.

© 2015 H. Sasaki, V. Tangkaratt & M. Sugiyama.

A seminal work on SDR is *sliced inverse regression* (SIR) (Li, 1991), which tries to find the projection matrix based on the inverse regression function from output to input. Since then, a number of methods related to SIR have been proposed such as the *principal Hessian direction* (Li, 1992) and *sliced average variance estimation* (Cook, 2000). However, these methods rely on the strong assumption that the underlying density of input data is elliptic, which can be easily violated in practice.

An alternative method applicable to a wider class of data is *minimum average variance estimation based on conditional density functions* (dMAVE) (Xia, 2007), which uses the conditional density of the output given the projected input for finding a projection matrix. dMAVE non-parametrically estimates the conditional density by a *local linear smoother* (Fan and Gijbels, 1996), and has been proved to find the correct projection matrix asymptotically. However, the local linear smoother tends to perform poorly when the input dimensionality is high, and the iterative estimation procedure of dMAVE is computationally demanding for large datasets. In addition, it is usually difficult to tune parameters such as the bandwidth in the local linear smoother. Therefore, the practical usage of dMAVE seems to be limited.

As a SDR method equipped with reliable parameter tuning, least-squares dimension reduction (LSDR) has been proposed (Suzuki and Sugiyama, 2013). LSDR is an information-theoretic SDR method involving non-parametric estimation of a squared-loss variant of *mutual information*. Unlike dMAVE, a notable advantage of LSDR is that it is equipped with a systematic parameter tuning method. However, the drawback of LSDR is that the estimation of a projection matrix requires to solve a non-convex optimization problem, which suffers from multiple local optima in general. Multiple starts with different initialization can mitigate this drawback, but this in return increases the computation costs significantly.

An attractive and computationally efficient approach is based on the *gradients of regression functions* (Samarov, 1993; Hristache et al., 2001; Xia et al., 2002). In stark contrast to dMAVE and LSDR, the advantage is that this approach does not require any iterative optimization procedure, and a projection matrix can be obtained analytically via eigenvalue decomposition. However, this approach is not guaranteed to perform SDR, since it only tries to fulfill the necessity of SDR (Xia, 2007; Fukumizu and Leng, 2014). The insufficiency of the above gradient-based approach can be overcome by the *outer product of gradient based on the conditional density functions* (dOPG) (Xia, 2007), which employs the gradient of the conditional density of output given input. However, since a local linear smoother is used to estimate the gradient, dOPG essentially shares the same drawbacks as dMAVE.

To extend the applicability of the gradient-based approach, in this paper, we propose a novel SDR method based on a new estimator for the gradients of logarithmic conditional densities called the *least-squares logarithmic conditional density gradients* (LSLCG). The fundamental idea of LSLCG is to *directly* fit a log-gradient model to the true log-gradient under the squared loss. This simple formulation yields a closed-form solution that can be computed efficiently. In addition, all tuning parameters included in LSLCG can be objectively determined by cross-validation. Through experiments, we demonstrate that the SDR method based on LSLCG outperforms existing SDR methods both in estimation accuracy and computational efficiency.

This paper is organized as follows: In Section 2, we first formulate the problem of SDR, and then we review existing methods and discuss their weaknesses. Section 3 gives LSLCG, a novel estimator for the gradients of logarithmic conditional densities, and then proposes the SDR algorithm based

on LSLCG. In Section 4, we experimentally evaluate the performance of the LSLCG-based SDR method. Finally, Section 5 concludes this paper.

2. Review of Existing Methods

In this section, the problem of SDR is formulated, and then existing SDR methods are reviewed.

2.1. Problem Formulation

Suppose that we are given a collection of input and output data,

$$\mathcal{D} = \left\{ (y_i, \boldsymbol{x}_i) \;\middle|\; y_i \in \mathbb{R}, \boldsymbol{x}_i = (x_i^{(1)}, \ldots, x_i^{(d_{\mathbf{x}})}) \in \mathbb{R}^{d_{\mathbf{x}}} \right\}_{i=1}^{n},$$

which are drawn independently from the joint distribution with density $p(y, \boldsymbol{x})$. Further, we assume that the input dimensionality $d_{\mathbf{x}}$ is large, but the "intrinsic" dimensionality of $\boldsymbol{x}$, which is denoted by $d_{\mathbf{z}}$, is rather small, and $\boldsymbol{z} \in \mathbb{R}^{d_{\mathbf{z}}}$ is the orthogonal projection of input data $\boldsymbol{x}$ with a low-rank matrix $\mathbf{B} \in \mathbb{R}^{d_{\mathbf{x}} \times d_{\mathbf{z}}}$:

$$\boldsymbol{z} = \mathbf{B}^\top \boldsymbol{x},$$

where $\mathbf{B}^\top \mathbf{B} = \mathbf{I}_{d_{\mathbf{z}}}$ and $\mathbf{I}_{d_{\mathbf{z}}}$ is the $d_{\mathbf{z}}$ by $d_{\mathbf{z}}$ identity matrix. The goal of SDR is to estimate $\mathbf{B}^*$ from $\mathcal{D}$ such that the following SDR condition is satisfied:

$$p(y|\boldsymbol{x}) = p(y|\mathbf{B}^{*\top}\boldsymbol{x}). \tag{1}$$

Throughout this paper, we denote the subspace spanned by the columns of $\mathbf{B}$ by Span$(\mathbf{B})$.

2.2. Existing Methods for Sufficient Dimension Reduction

Here, we review some existing SDR methods and discuss their weaknesses. For other SDR methods, the reader may refer to Adragni and Cook (2009), Burges (2009) and Ma and Zhu (2013).

2.2.1. MINIMUM AVERAGE VARIANCE ESTIMATION BASED ON CONDITIONAL DENSITY FUNCTIONS (DMAVE)

A method applicable to a wide class of data is dMAVE (Xia, 2007). dMAVE non-parametrically estimates the conditional density $p(y|\mathbf{B}^\top \boldsymbol{x})$ through a local linear smoother (LLS) (Fan and Gijbels, 1996). Consider the following regression-like model:

$$H_{b_H}(Y - y) = f(y, \mathbf{B}^\top \boldsymbol{x}) + \epsilon,$$

where $f(y, \mathbf{B}^\top \boldsymbol{x})$ is a model, and $H_{b_H}(Y - y)$ denotes a symmetric unimodal kernel with the bandwidth parameter b_H. Assuming that the conditional expectation of ϵ given the projected data with respect to Y is zero, i.e. $E_Y\{\epsilon|\mathbf{B}^\top \boldsymbol{x}\} = 0$, we obtain

$$E_Y\{H_{b_H}(Y - y)|\mathbf{B}^\top \boldsymbol{x}\} = f(y, \mathbf{B}^\top \boldsymbol{x}).$$

The key point of the above equation is that when $b_H \to 0$ as $n \to \infty$, the conditional expectation $E_Y\{H_{b_H}(Y - y)|\mathbf{B}^\top \boldsymbol{x}\}$ approaches to the conditional density $p(y|\mathbf{B}^\top \boldsymbol{x})$. Thus, estimating $f(y, \mathbf{B}^\top \boldsymbol{x})$ is asymptotically equivalent to estimating $p(y|\mathbf{B}^\top \boldsymbol{x})$.

To estimate $f(y, \mathbf{B}^\top \boldsymbol{x})$ and $\mathbf{B}$, the first-order Taylor approximation is applied as follows (Fan and Gijbels, 1996):

$$f(y, \mathbf{B}^\top \boldsymbol{x}) \approx f(y, \mathbf{B}^\top \boldsymbol{x}_i) + (\nabla_{\boldsymbol{z}} f(y, \boldsymbol{z})|_{\boldsymbol{z}=\mathbf{B}^\top \boldsymbol{x}_i})^\top \mathbf{B}^\top (\boldsymbol{x} - \boldsymbol{x}_i) \quad \text{for} \quad \mathbf{B}^\top \boldsymbol{x} \approx \mathbf{B}^\top \boldsymbol{x}_i, \tag{2}$$

where $\nabla_{\boldsymbol{z}}$ denotes the vector differential operator with respect to $\boldsymbol{z}$. Then, the optimization problem[1] is formulated as

$$\left(\widehat{\mathbf{B}}, \widehat{\gamma}'_{ki}, \widehat{\boldsymbol{\zeta}}'_{ki}\right) = \underset{(\mathbf{B}, \gamma'_{ki}, \boldsymbol{\zeta}'_{ki})}{\arg\min}$$
$$\frac{1}{n^3} \sum_{k=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} [H_{b_H}(y_k - y_i) - \underbrace{f(y_k, \mathbf{B}^\top \boldsymbol{x}_i)}_{\gamma'_{ki}} - \underbrace{\nabla_{\boldsymbol{z}}(f(y_k, \boldsymbol{z})|_{\boldsymbol{z}=\mathbf{B}^\top \boldsymbol{x}_i})^\top}_{(\boldsymbol{\zeta}'_{ki})^\top} \mathbf{B}^\top (\boldsymbol{x}_j - \boldsymbol{x}_i)]^2 w_{ij}, \tag{3}$$

where $w_{ij} = K_h\left(\|\mathbf{B}^\top (\boldsymbol{x}_j - \boldsymbol{x}_i)\|\right)$ is a weight function with a kernel function K_h containing the bandwidth parameter h. $\mathbf{B}$ and $(\gamma'_{kj}, \boldsymbol{\zeta}'_{kj})$ are alternately and iteratively updated.

Since dMAVE does not assume any forms to the conditional density $p(y|\mathbf{B}^\top \boldsymbol{x})$, it is applicable to a wide class of data. However, dMAVE has mainly three drawbacks:

1. In the Taylor approximation (2), the proximity requirement $\mathbf{B}^\top \boldsymbol{x}_j \approx \mathbf{B}^\top \boldsymbol{x}_i$ holds only when data is dense. Thus, dMAVE might not perform well when data is sparse in the subspace. This is in particular problematic when the dimensionality $d_{\mathbf{z}}$ is relatively high.

2. The alternate updates for $\mathbf{B}$ and $(\gamma'_{ki}, \boldsymbol{\zeta}'_{ki})$ in dMAVE can be computationally expensive for large datasets because the number of elements in $(\gamma'_{ki}, \boldsymbol{\zeta}'_{ki})$ is proportional to n^2.

3. The parameter values for b_H and h are determined based on the normal reference rule of the non-parametric conditional density (Silverman, 1986; Fan and Gijbels, 1996). Therefore, when the data density is far from the normal density, this parameter selection method may not work well.

2.2.2. LEAST-SQUARES DIMENSION REDUCTION (LSDR)

LSDR (Suzuki and Sugiyama, 2013) is an information-theoretic SDR method, which includes a reliable parameter selection method. LSDR is based on the squared-loss mutual information (SMI): SMI is an independence measure alternative to mutual information, which is defined by

$$\mathrm{SMI}(Y, \boldsymbol{X}) = \frac{1}{2} \int \left(\frac{p_{\mathrm{y}\mathbf{x}}(y, \boldsymbol{x})}{p_{\mathrm{y}}(y) p_{\mathbf{x}}(\boldsymbol{x})} - 1 \right)^2 p_{\mathrm{y}}(y) p_{\mathbf{x}}(\boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x},$$

where $p_{\mathrm{y}\mathbf{x}}(y, \boldsymbol{x})$ is the joint density, and $p_{\mathrm{y}}(y)$ and $p_{\mathbf{x}}(\boldsymbol{x})$ denote the marginal densities of y and $\boldsymbol{x}$, respectively. As proved in (Suzuki and Sugiyama, 2013), the rationale of using SMI for SDR comes from the following inequality:

$$\mathrm{SMI}(Y, \boldsymbol{X}) \geq \mathrm{SMI}(Y, \boldsymbol{Z}). \tag{4}$$

1. The actual formulation in dMAVE has a trimming function for technical purposes in front of the summation $\sum_{i=1}^{n}$ in (3). Here, for notational simplicity, we removed it.

Interestingly, the equality holds if and only if the SDR condition (1) is satisfied. Thus, maximizing SMI$(Y, \boldsymbol{Z})$ provides $\mathbf{B}^*$. When estimating SMI$(Y, \boldsymbol{Z})$ from data in LSDR, a non-parametric density-ratio estimator (Kanamori et al., 2009) has been applied. Then, estimation of SMI and maximization of the SMI estimator with respect to $\mathbf{B}$ are performed alternately.

Unlike dMAVE, a notable advantage of LSDR is that a cross-validation method is provided for tuning all the parameters. Furthermore, LSDR also does not have any assumptions on densities. However, the main drawback is that estimating $\mathbf{B}$ involves solving a non-convex optimization problem, which suffers from multiple local optima in general. To cope with this problem, multiple point search with different initial values on $\mathbf{B}$ may be performed. However, this procedure is computationally inefficient. Recently, another information-theoretic method has been proposed in (Tangkaratt et al., 2015). However, this method also needs to solve a non-convex optimization problem when estimating $\mathbf{B}$.

2.2.3. ESTIMATION BASED ON OUTER PRODUCTS OF GRADIENTS

A computationally efficient approach in contrast to dMAVE and LSDR is based on the gradients of regression functions (Samarov, 1993; Hristache et al., 2001; Xia et al., 2002). The reason of using the gradients is due to the following equation which can be easily derived from the SDR condition (1):

$$\begin{aligned}\nabla_{\boldsymbol{x}} E\{y|\boldsymbol{x}\} &= \mathbf{B}^* \nabla_{\boldsymbol{z}^*} E\{y|\boldsymbol{z}^*\} \\ &= c_1 \boldsymbol{b}_1^* + c_2 \boldsymbol{b}_2^* + \cdots + c_{d_{\mathbf{z}}} \boldsymbol{b}_{d_{\mathbf{z}}}^*, \end{aligned} \tag{5}$$

where $\boldsymbol{z}^* = \mathbf{B}^{*\top} \boldsymbol{x}$, c_k is the partial derivative of $E\{y|\boldsymbol{z}^*\}$ with respect to the k-th coordinate in $\boldsymbol{z}^*$, and $\boldsymbol{b}_k^*$ denotes the k-th column vector in $\mathbf{B}^*$. Eq.(5) indicates that the gradient $\nabla_{\boldsymbol{x}} E\{y|\boldsymbol{x}\}$ is contained in Span$(\mathbf{B}^*)$. Therefore, $\mathbf{B}^*$ can be estimated as a collection of the top $d_{\mathbf{z}}$ eigenvectors from the expectation of the outer products of the estimates to $\nabla_{\boldsymbol{x}} E\{y|\boldsymbol{x}\}$. Since any iterative optimization steps and solving non-convex optimization problems are not needed, this approach is computationally efficient. However, (5) only satisfies the necessary condition of (1), and the sufficient condition is not satisfied in general. For example, when y follows a regression model $y = x^{(1)} + x^{(2)}\epsilon'$ where $E\{\epsilon'|\boldsymbol{x}\} = 0$, the regression function $E\{y|\boldsymbol{x}\}$ depends on only $x^{(1)}$, but the conditional density $p(y|\boldsymbol{x})$ depends on both $x^{(1)}$ and $x^{(2)}$ (Fukumizu and Leng, 2014, Section 2.1).

To extend the applicability of this approach, recently, a method called *gradient-based kernel dimension reduction* (gKDR) has been proposed based on the gradient of a conditional expectation $E\{g(y)|\boldsymbol{x}\}$ where g is a function in a reproducing kernel Hilbert space (Fukumizu and Leng, 2014). Based on the kernel method, gKDR estimates $\mathbf{B}^*$ by applying the eigenvalue decomposition to the expectation of

$$\nabla_{\boldsymbol{x}} k_{\mathbf{x}}(\boldsymbol{x})(\mathbf{G}_{\boldsymbol{X}} + n\epsilon_n \mathbf{I}_n)^{-1} \mathbf{G}_{\boldsymbol{Y}} (\mathbf{G}_{\boldsymbol{X}} + n\epsilon_n \mathbf{I}_n)^{-1} \nabla_{\boldsymbol{x}} k_{\mathbf{x}}(\boldsymbol{x})^\top, \tag{6}$$

where $\mathbf{G}_{\boldsymbol{X}}$ and $\mathbf{G}_{\boldsymbol{Y}}$ are Gram matrices whose elements are kernel values $k_{\mathbf{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $k_{\mathrm{y}}(y_i, y_j)$ respectively, $\nabla_{\boldsymbol{x}} k_{\mathbf{x}}(\boldsymbol{x}) = (\nabla_{\boldsymbol{x}} k_{\mathbf{x}}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \nabla_{\boldsymbol{x}} k_{\mathbf{x}}(\boldsymbol{x}, \boldsymbol{x}_n)) \in \mathbb{R}^{d_{\mathbf{x}} \times n}$, and ϵ_n denotes the regularization parameter. However, the performance of gKDR depends on the parameter ϵ_n and the kernel parameters in $k_{\mathbf{x}}$ and k_{y}, and there seems to be no systematic model selection method for dimension reduction based on kernel methods (Fukumizu et al., 2004, 2009)[2].

2. In principle, model selection can be performed by cross-validation (CV) over a successive predictor. However, this should be avoided in practice because of the two reasons: First, when CV is applied, one should optimize both

An alternative solution to the problem based on the gradients of regression functions is to use gradient estimates of the conditional density $p(y|\boldsymbol{x})$. As in (5), it can be easily proved that the gradient of the conditional density is also contained in Span$(\mathbf{B}^*)$ as

$$\nabla_{\boldsymbol{x}} p(y|\boldsymbol{x}) = \mathbf{B}^* \nabla_{\boldsymbol{z}^*} p(y|\boldsymbol{z}^*). \tag{7}$$

Unlike the approach based on regression functions, this approach satisfies the sufficient condition as well, which can be proved similarly in Fukumizu and Leng (2014, Theorem 2). *The outer product of gradient based on the conditional density functions* (dOPG) non-parametrically estimates the gradient by LLS (Xia, 2007). As described in Section 2.2.1, by considering a regression-like model,

$$H_{b_H}(Y - y) = m(y, \boldsymbol{x}) + \epsilon,$$

where $m(y, \boldsymbol{x})$ is a model and $E_Y\{\epsilon|\boldsymbol{x}\} = 0$, estimating $m(y, \boldsymbol{x})$ is asymptotically equivalent to estimating the conditional density $p(y|\boldsymbol{x})$. Then, $m(y, \boldsymbol{x})$ and its gradient $\nabla_{\boldsymbol{x}} m(y, \boldsymbol{x})$ are estimated with the first-order Taylor approximation as the minimizers of

$$\left(\widehat{\gamma}_{ki}, \widehat{\boldsymbol{\zeta}}_{ki}\right) = \underset{(\gamma_{ki}, \boldsymbol{\zeta}_{ki})}{\arg\min} \sum_{i=1}^{n} [H_{b_H}(y_k - y_i) - \underbrace{m(y_k, \boldsymbol{x}_i)}_{\gamma_{ki}} - \underbrace{\nabla_{\boldsymbol{x}}(m(y_k, \boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_i})^\top}_{\boldsymbol{\zeta}_{ki}^\top}(\boldsymbol{x}_j - \boldsymbol{x}_i)]^2 w_{ij}, \tag{8}$$

where $w_{ij} = K_h(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|)$. Then, the projection matrix $\mathbf{B}$ is estimated through the eigenvalue decomposition. A dOPG algorithm is summarized in Algorithm 1.[3] However, the practical applicability of dOPG seems to be limited because dOPG employs LLS to estimate the gradient and thus shares the same drawbacks as dMAVE.

To cope with the problems of LLS in gradient estimation, we derive a novel estimator for the gradients of logarithmic conditional densities and propose a SDR method based on the estimator.

3. Least-Squares Logarithmic Conditional Density Gradients

In this section, we propose a novel estimator for the gradients of logarithmic conditional densities:

$$\partial_j \log p(y|\boldsymbol{x}) = \frac{\partial_j p(y|\boldsymbol{x})}{p(y|\boldsymbol{x})} = \frac{\partial_j p(y, \boldsymbol{x}) p(\boldsymbol{x}) - p(y, \boldsymbol{x}) \partial_j p(\boldsymbol{x})}{p(y, \boldsymbol{x}) p(\boldsymbol{x})} = \frac{\partial_j p(y, \boldsymbol{x})}{p(y, \boldsymbol{x})} - \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})}, \tag{9}$$

where $\partial_j = \frac{\partial}{\partial x^{(j)}}$. As shown below, the proposed estimator does not have any requirement unlike LLS, and analytically computes the solutions. A cross-validation method for parameter tuning is also provided.

parameters in a SDR method and hyper-parameters in the predictor. This procedure results in a nested CV, which is computationally quite inefficient. Second, features extracted based on CV are no longer independent of predictors, which is not preferable in terms of interpretability (Suzuki and Sugiyama, 2013).

3. The actual dOPG algorithm proposed in Xia (2007, Section 2.1) is slightly different from Algorithm 1, and uses some techniques to improve estimation. Here, we presented a simpler algorithm to make clearer the fundamental difference from our algorithm.

Input: Data samples, $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$.

1. Estimate the gradient of the conditional density for y_k and $\boldsymbol{x}_i$ as $\widehat{\boldsymbol{\zeta}}_{ki}$.
2. Compute $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n^2}\sum_{k=1}^n \sum_{i=1}^n \widehat{\boldsymbol{\zeta}}_{ki}\widehat{\boldsymbol{\zeta}}_{ki}^\top$.
3. Obtain the $d_{\mathbf{z}}$ eigenvectors, $\boldsymbol{e}_1, \boldsymbol{e}_2, \dots \boldsymbol{e}_{d_{\mathbf{z}}}$, via eigenvalue decomposition of $\widehat{\boldsymbol{\Sigma}}$.

Output: $\widehat{\mathbf{B}} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \dots \boldsymbol{e}_{d_{\mathbf{z}}})$.

Figure 1: An algorithm of dOPG. The $d_{\mathbf{z}}$ eigenvectors in the algorithm mean the eigenvectors corresponding to the largest $d_{\mathbf{z}}$ eigenvalues of $\widehat{\boldsymbol{\Sigma}}$.

3.1. The Estimator

The fundamental idea is to fit a gradient model, $\boldsymbol{g}(y, \boldsymbol{x}) = (g^{(1)}(y, \boldsymbol{x}), \dots, g^{(d_{\mathbf{x}})}(y, \boldsymbol{x}))^\top$, *directly* to the true gradient of the logarithmic conditional density of y given $\boldsymbol{x}$ under the squared loss:

$$
\begin{aligned}
J(g^{(j)}) &= \iint \left\{ g^{(j)}(y, \boldsymbol{x}) - \partial_j \log p(y|\boldsymbol{x}) \right\}^2 p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} - C_j \\
&= \iint \left\{ g^{(j)}(y, \boldsymbol{x}) \right\}^2 p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} - 2 \iint g^{(j)}(y, \boldsymbol{x}) \partial_j p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} \\
&\quad + 2 \iint g^{(j)}(y, \boldsymbol{x}) \left\{ \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\} p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} \\
&= \iint \left\{ g^{(j)}(y, \boldsymbol{x}) \right\}^2 p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} - 2 \iint \left[g^{(j)}(y, \boldsymbol{x}) p(y, \boldsymbol{x}) \right]_{x^{(j)}=-\infty}^{x^{(j)}=\infty} \mathrm{d}y \mathrm{d}\boldsymbol{x}_{\setminus x^{(j)}} \\
&\quad + 2 \iint \partial_j g^{(j)}(y, \boldsymbol{x}) p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} + 2 \iint g^{(j)}(y, \boldsymbol{x}) \left\{ \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\} p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} \\
&= \iint \left\{ g^{(j)}(y, \boldsymbol{x}) \right\}^2 p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} \\
&\quad + 2 \iint \partial_j g^{(j)}(y, \boldsymbol{x}) p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} + 2 \iint g^{(j)}(y, \boldsymbol{x}) \left\{ \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\} p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x},
\end{aligned}
$$

where $C_j = \iint \{\partial_j \log p(y|\boldsymbol{x})\}^2 p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x}$, $\mathrm{d}\boldsymbol{x}_{\setminus x^{(j)}}$ denotes the integration except for $x^{(j)}$, and we reached the last equality by applying *integration by parts* to the second term in the second line under a mild assumption that $\lim_{|x^{(j)}|\to\infty} g^{(j)}(y, \boldsymbol{x}) p(y, \boldsymbol{x}) = 0$. We can easily derive the empirical approximation of the first and second terms in the last equality, but it is difficult to empirically approximate the third term because it includes the true partial derivative of the log-density, $\partial_j \log p(\boldsymbol{x}) = \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})}$. To cope with this problem, we employ a non-parametric plug-in estimator for log-density gradients, which we call the *least-squares log-density gradients* (LSLDG) (Cox, 1985; Sasaki et al., 2014). The estimation accuracy of LSLDG has been shown to be much better than that of a log-density-gradient estimator based on kernel density estimation particularly for high-dimensional data, and LSLDG efficiently computes all the solutions in a closed form. Furthermore, all the tuning parameters in LSLDG can be cross-validated. Thus, LSLDG would provide an accurate and efficient approximation of the third term.

Substituting $\frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})}$ by the LSLDG estimator $\widehat{r}^{(j)}(\boldsymbol{x})$, the loss $J(g^{(j)})$ is approximated as

$$J(g^{(j)}) \approx \iint \left\{g^{(j)}(y,\boldsymbol{x})\right\}^2 p(y,\boldsymbol{x})\mathrm{d}y\mathrm{d}\boldsymbol{x} + 2\iint \left\{\partial_j g^{(j)}(y,\boldsymbol{x}) + g^{(j)}(y,\boldsymbol{x})\widehat{r}^{(j)}(\boldsymbol{x})\right\} p(y,\boldsymbol{x})\mathrm{d}y\mathrm{d}\boldsymbol{x}.$$

Then, the empirical version of the approximative loss is given by

$$\tilde{J}(g^{(j)}) = \frac{1}{n}\sum_{i=1}^{n}\left\{g^{(j)}(y_i,\boldsymbol{x}_i)\right\}^2 + 2\left\{\partial_j g^{(j)}(y_i,\boldsymbol{x}_i) + g^{(j)}(y_i,\boldsymbol{x}_i)\widehat{r}^{(j)}(\boldsymbol{x}_i)\right\}. \tag{10}$$

To estimate $\partial_j \log p(y|\boldsymbol{x})$, we use the following linear-in-parameter model:

$$g^{(j)}(y,\boldsymbol{x}) = \sum_{i=1}^{n}\theta_{ij}\underbrace{\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2(\sigma_x^{(j)})^2} - \frac{(y-y_i)^2}{2\sigma_y^2}\right)}_{\psi_j^{(i)}(y,\boldsymbol{x})} = \boldsymbol{\theta}_j^\top \boldsymbol{\psi}_j(y,\boldsymbol{x}),$$

$$\partial_j g^{(j)}(y,\boldsymbol{x}) = \sum_{i=1}^{n}\theta_{ij}\underbrace{\partial_j \psi_j^{(i)}(y,\boldsymbol{x})}_{\phi_j^{(i)}(y,\boldsymbol{x})} = \boldsymbol{\theta}_j^\top \boldsymbol{\phi}_j(y,\boldsymbol{x}),$$

where $\psi_j^{(i)}(y,\boldsymbol{x})$ is a basis function.[4] By substituting this model and adding the ℓ_2 regularizer to (10), we obtain the closed-form solution as

$$\widehat{\boldsymbol{\theta}}_j = \arg\min_{\boldsymbol{\theta}_j}\left[\boldsymbol{\theta}_j^\top \mathbf{G}_j \boldsymbol{\theta}_j + 2\boldsymbol{\theta}_j^\top \boldsymbol{h}_j + \lambda^{(j)}\boldsymbol{\theta}_j^\top \boldsymbol{\theta}_j\right] = -(\mathbf{G}_j + \lambda^{(j)}\mathbf{I})^{-1}\boldsymbol{h}_j,$$

where $\lambda^{(j)} \geq 0$ is the regularization parameter,

$$\mathbf{G}_j = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\psi}_j(y_i,\boldsymbol{x}_i)\boldsymbol{\psi}_j(y_i,\boldsymbol{x}_i)^\top \text{ and } \boldsymbol{h}_j = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\phi}_j(\boldsymbol{x}_i) + \widehat{r}^{(j)}(\boldsymbol{x}_i)\boldsymbol{\psi}_j(\boldsymbol{x}_i).$$

Finally, the estimator, which we call the *least-squares logarithmic conditional density gradients* (LSLCG), is obtained as

$$\widehat{g}^{(j)}(\boldsymbol{x}) = \widehat{\boldsymbol{\theta}}_j^\top \boldsymbol{\psi}_j(\boldsymbol{x}).$$

3.2. Model Selection by Cross-Validation

The performance of LSLCG depends on the choice of models, which are the Gaussian width parameters, $\sigma_x^{(j)}$ and σ_y, and the regularization parameter $\lambda^{(j)}$ in the current setup. We perform model selection by cross-validation as follows:

Step 1 Divide the sample $\mathcal{D} = \{(y_i,\boldsymbol{x}_i)\}_{i=1}^n$ into T disjoint subsets $\{\mathcal{D}_t\}_{t=1}^T$.

4. When n is too large, we may only use a subset of data samples as center points for computational efficiency.

Step 2 Obtain an estimator $\widehat{g}_t^{(j)}(\boldsymbol{x})$ using $\mathcal{D} \setminus \mathcal{D}_t$, and then compute the hold-out error to $\mathcal{D}_t$ as

$$\mathrm{CV}(t) = \frac{1}{|\mathcal{D}_t|} \sum_{(y,\boldsymbol{x}) \in \mathcal{D}_t} \left\{ \widehat{g}_t^{(j)}(y, \boldsymbol{x}) \right\}^2 + 2 \left\{ \partial_j \widehat{g}_t^{(j)}(y, \boldsymbol{x}) + \widehat{g}_t^{(j)}(y, \boldsymbol{x}) \widehat{r}^{(j)}(\boldsymbol{x}) \right\}, \tag{11}$$

where $|\mathcal{D}_t|$ denotes the number of elements in $\mathcal{D}_t$.

Step 3 Choose the model that minimizes $\mathrm{CV} = \frac{1}{T} \sum_{t=1}^{T} \mathrm{CV}(t)$.

3.3. Least-Squares Gradients for Dimension Reduction

Here, we propose our SDR method. Our algorithm is essentially the same as the one in Figure 1: We replace the gradient estimates from LLS to LSLCG, and perform eigenvalue decomposition to $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} \widehat{\boldsymbol{g}}(y_i, \boldsymbol{x}_i) \widehat{\boldsymbol{g}}(y_i, \boldsymbol{x}_i)^\top$, where $\widehat{\boldsymbol{g}}(y_i, \boldsymbol{x}_i)$ is an estimate from LSLCG. We call this method the *least-squares gradients for dimension reduction* (LSGDR). A MATLAB package is available from https://sites.google.com/site/hworksites/home/software/lsgdr.

4. Numerical Experiments for Sufficient Dimension Reduction

In this section, we investigate the performance of the proposed SDR method using both artificial and benchmark datasets, and compare it with the existing methods.

4.1. Illustration of Dimension Reduction on Artificial Data

First, we illustrate the behavior of LSGDR using artificial data, and the comparison to dMAVE[5], LSDR[6] and gKDR[7].

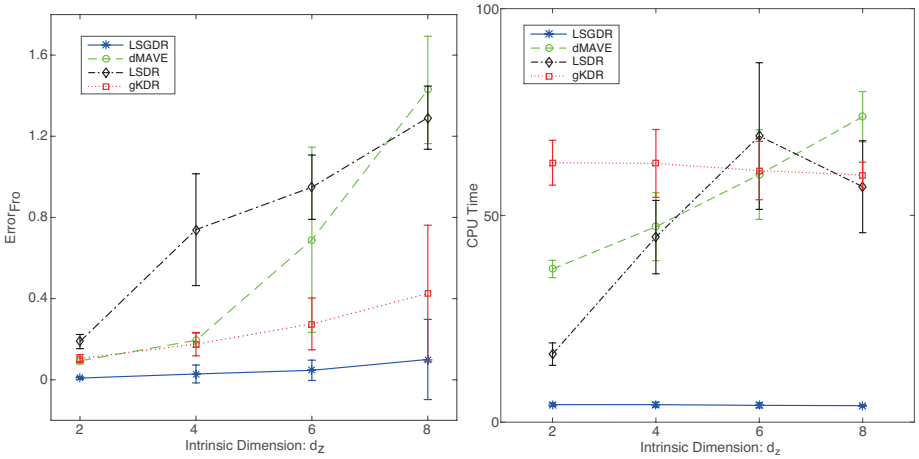
4.1.1. INTRINSIC-DIMENSION-SCALABILITY AND COMPUTATIONAL EFFICIENCY

To investigate the scalability to the intrinsic-dimension $d_{\mathbf{z}}$ and computational efficiency, we generated the outputs y according to the model,

$$y = \sum_{j=1}^{d_{\mathbf{z}}} (x^{(j)})^2 + 0.3\epsilon,$$

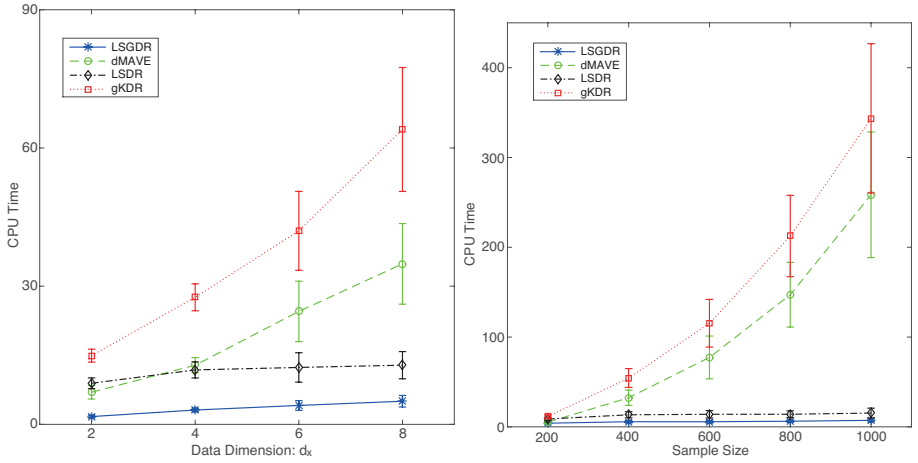
where $\boldsymbol{x}$ and ϵ were drawn from the standard normal density. The data dimension and total number of samples were $d_{\mathbf{x}} = 10$ and $n = 500$, respectively. After generating data, y and $\boldsymbol{x}$ were standardized in a element-wise manner to have zero means and unit variances. The estimation error was assessed by $\mathrm{Error}_{\mathrm{Fro}} = \|\widehat{\mathbf{B}}\widehat{\mathbf{B}}^\top - \mathbf{B}^*\mathbf{B}^{*\top}\|_{\mathrm{Fro}}$, where $\|\cdot\|_{\mathrm{Fro}}$ denotes the Frobenius norm, $\mathbf{B}^*$ is the true projection matrix, and $\widehat{\mathbf{B}}$ is its estimate. For LSGDR, we only used 100 center points in the basis functions $\psi_j^{(i)}(\boldsymbol{x})$ which were randomly selected from data samples. $\sigma_x^{(j)}$ and $\lambda^{(j)}$ are cross-validated as in Section 3.2.[8]

5. http://www.stat.nus.edu.sg/~staxyc/dMAVE.m
6. http://www.ms.k.u-tokyo.ac.jp/software.html#LSDR
7. http://www.ism.ac.jp/~fukumizu/software.html
8. To reduce the computation cost, we fix σ_y as the median value of $|y_i - y_j|$ for all i and j throughout the remaining parts of this paper.



(*a*) Error against intrinsic dimension $d_{\mathbf{z}}$ ($d_{\mathbf{x}} = 10, n = 500$).

(*b*) CPU time against intrinsic dimension $d_{\mathbf{z}}$ ($d_{\mathbf{x}} = 10, n = 500$).



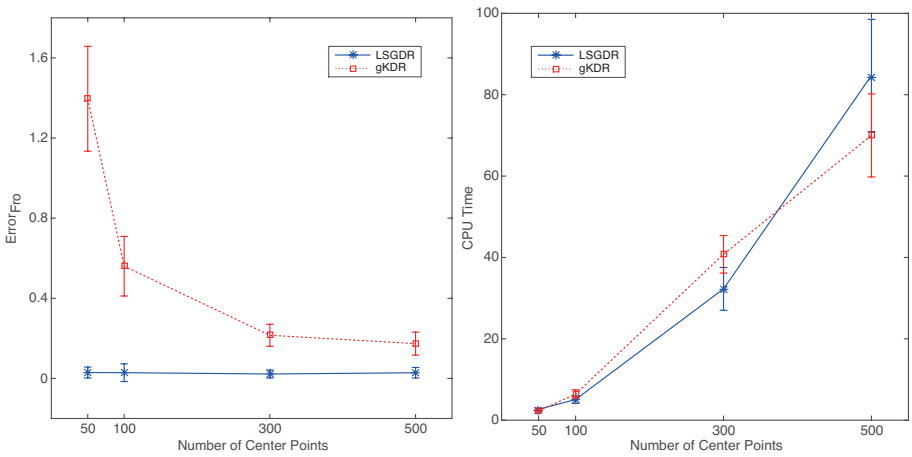
(*c*) CPU time against original data dimension $d_{\mathbf{x}}$ ($d_{\mathbf{z}} = 1, n = 500$).

(*d*) CPU time against sample size n ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 1$).

Figure 2: Comparison to existing methods in terms of (a) the estimation error and (b,c,d) computation cost. Each point denotes the mean value over 50 runs, and the error bars are standard deviations.

Figure 2(a) shows that LSGDR produces the best performance and gKDR works reasonably well to a wide range of the intrinsic dimensionality $d_{\mathbf{z}}$, while the estimation errors in dMAVE and LSDR sharply increase as $d_{\mathbf{z}}$ increases. A reason of this unsatisfactory performance for LSDR is that good initialization on $\mathbf{B}$ is more challenging for data contained in a higher-dimensional subspace. For dMAVE, LLS seems not to work well to data with large $d_{\mathbf{z}}$ as reviewed in Section 2.2.1.

Figure 2(b) reveals that LSGDR is computationally the most efficient method. Since dMAVE and LSDR estimate the projection matrix $\mathbf{B}$ using the data projected on the subspace, the computation costs of these methods increase as the intrinsic-dimensionality $d_{\mathbf{z}}$ grows. On the other hand, changing $d_{\mathbf{z}}$ does not affect the computation costs of gKDR and LSGDR which use (non-projected)



(*a*) Error against the number of center points ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 4, n = 500$).

(*b*) CPU time against the number of center points ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 4, n = 500$).

Figure 3: Comparison to gKDR when the number of center points is changed.

data. Instead, when data dimension $d_{\mathbf{x}}$ increases, the computation costs of gKDR and LSGDR grow (Figure 2(c)). Note that the computation cost of LSGDR increases more mildly and is still the smallest among all the methods. This is because LSGDR uses only 100 centers in the basis function, which is independent of the sample size n. The computation cost of dMAVE also increases with $d_{\mathbf{x}}$ because an estimate from dOPG is used for initialization on $\mathbf{B}$. Increasing the sample size n strongly grows the computation costs of dMAVE and gKDR (Figure 2(d)). For dMAVE, the number of elements $(\gamma'_{kj}, \boldsymbol{\zeta}'_{kj})$ in (3) is proportional to n^2, and gKDR has to compute the inverse of an $n \times n$ matrix in (6). For gKDR, the computation costs can be decreased by reducing the size of the Gram matrices, $\mathbf{G}_{\mathbf{X}}$ and $\mathbf{G}_{\mathbf{Y}}$, and the number of centers $\boldsymbol{x}_i$ in the kernel function $k_{\mathbf{x}}(\boldsymbol{x}, \boldsymbol{x}_i)$ as done in LSGDR. However, this is not a good approach in gKDR: Using a smaller number of center points reduces the computation costs, but significantly increases the estimation error (Figure 3(a) and (b)). In contrast to gKDR, LSGDR keeps the estimation error small against fewer center points, while the computation costs are reduced dramatically.

4.1.2. PERFORMANCE ON VARIOUS KINDS OF ARTIFICIAL DATA

Here, we generated data according to the following various kinds of models, all of which were adopted from the papers of dMAVE, LSDR and gKDR:

(a) (Xia, 2007) ($d_{\mathbf{x}} = 20, d_{\mathbf{z}} = 2, n = 200$)

$$y = \mathrm{sgn}(2\boldsymbol{x}^\top \boldsymbol{\beta}^{(1)} + \epsilon^{(1)}) \log(|2\boldsymbol{x}^\top \boldsymbol{\beta}^{(2)} + 4 + \epsilon^{(2)}|),$$

where $\mathrm{sgn}(\cdot)$ denotes the sign function, $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_{\mathbf{x}}})$, $\epsilon^{(1)}, \epsilon^{(2)} \sim \mathcal{N}(0, 1)$, and $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ denotes the normal density with the mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$. The first four elements in $\boldsymbol{\beta}^{(1)}$ are all 0.5 while the others are zeros. For $\boldsymbol{\beta}^{(2)}$, the first four elements are 0.5, -0.5, 0.5 and -0.5, respectively, and the others are zeros.

Table 1: The means and standard deviations of estimation errors and CPU time over 50 runs. The numbers in the parentheses are standard deviations. The best and comparable methods judged by the Wilcoxon signed-rank test at the significance level 1% are described in boldface.

(a) (Xia, 2007) ($d_{\mathbf{x}} = 20, d_{\mathbf{z}} = 2, n = 200$)

	LSGDR	dMAVE	LSDR	gKDR
$\text{Error}_{\text{Fro}}$	**0.378(0.112)**	0.643(0.166)	0.656(0.103)	0.784(0.165)
Time	**4.685(0.239)**	15.244(0.750)	9.392(1.877)	23.668(0.642)

(b) (Xia, 2007) ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 2, n = 500$)

	LSGDR	dMAVE	LSDR	gKDR
$\text{Error}_{\text{Fro}}$	**0.305(0.099)**	**0.280(0.059)**	0.390(0.085)	0.490(0.172)
Time	**3.218(0.308)**	24.152(2.980)	8.552(1.271)	50.753(4.292)

(c) (Suzuki and Sugiyama, 2013) ($d_{\mathbf{x}} = 4, d_{\mathbf{z}} = 2, n = 100$)

	LSGDR	dMAVE	LSDR	gKDR
$\text{Error}_{\text{Fro}}$	**0.104(0.114)**	0.225(0.120)	0.431(0.256)	0.340(0.209)
Time	1.213(0.163)	**0.205(0.048)**	4.343(0.560)	0.971(0.131)

(d) (Fukumizu and Leng, 2014) ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 2, n = 400$)

	LSGDR	dMAVE	LSDR	gKDR
$\text{Error}_{\text{Fro}}$	**0.286(0.471)**	1.018(0.342)	0.750(0.323)	0.791(0.366)
Time	**2.894(0.178)**	14.413(1.100)	14.606(2.473)	30.371(0.937)

(e) (Fukumizu and Leng, 2014) ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 1, n = 400$)

	LSGDR	dMAVE	LSDR	gKDR
$\text{Error}_{\text{Fro}}$	**0.043(0.029)**	0.150(0.043)	0.234(0.072)	0.220(0.070)
Time	**3.474(0.207)**	17.276(2.130)	6.016(0.605)	34.575(1.586)

(b) (Xia, 2007) ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 2, n = 500$)

$$y = 2\boldsymbol{x}^\top \boldsymbol{\beta}^{(1)} + 2\exp(\boldsymbol{x}^\top \boldsymbol{\beta}^{(2)})\epsilon,$$

where $\boldsymbol{x} \sim \text{Uni}[-\sqrt{3}, \sqrt{3}]^{d_{\mathbf{x}}}$, $\epsilon \sim \mathcal{N}(0,1)$, $\boldsymbol{\beta}^{(1)} = (1,2,0,0,0,0,0,0,0,2)^\top/3$, $\boldsymbol{\beta}^{(2)} = (0,0,3,4,0,0,0,0,0,0)^\top/5$, and $\text{Uni}[a,b]$ denotes the uniform density on $[a,b]$.

(c) (Suzuki and Sugiyama, 2013; Fukumizu et al., 2009) ($d_{\mathbf{x}} = 4, d_{\mathbf{z}} = 2, n = 100$)

$$y = \frac{x^{(1)}}{0.5 + (x^{(2)} + 1.5)^2} + (1.0 + x^{(2)})^2 + 0.4\epsilon,$$

where $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_{\mathbf{x}}})$ and $\epsilon \sim \mathcal{N}(0,1)$.

(d) (Fukumizu and Leng, 2014) ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 2, n = 400$)

$$y = ((x')^3 + x'')(x' - (x'')^3) + \epsilon,$$

where $x' = (x^{(1)} + x^{(2)})/\sqrt{2}$, $x'' = (x^{(1)} - x^{(2)})/\sqrt{2}$, $\boldsymbol{x} \sim \mathrm{Uni}[-1,1]^{d_{\mathbf{x}}}$, $\epsilon \sim \mathrm{Gamma}(1,2)$, and $\mathrm{Gamma}(k,\theta)$ denotes the Gamma density with the shape parameter k and scale parameter θ.

(e) (Fukumizu and Leng, 2014) ($d_{\mathbf{x}} = 10, d_{\mathbf{z}} = 1, n = 400$)

$$y = (x^{(1)} - 0.5)^4 \epsilon,$$

where $\boldsymbol{x}$ is drawn from a normal density $\mathcal{N}(0, 0.25)$ truncated on $[-1,1]$, and $\epsilon \sim \mathcal{N}(0,1)$.

The results are summarized in Table 1. Table 1 (a) and (b) show that the performance of LSGDR is best or comparable to the best method in terms of the estimation error, and computationally the most efficient. Since the models (a) and (b) have quite complex forms, these results ensure that LSGDR is a promising method to various kinds of data. For Table 1 (c), LSGDR is the most accurate method, but dMAVE has the smallest computation costs. Since dMAVE does not perform cross-validation for tuning the parameters unlike the other methods, it should be computationally efficient when n is small. However, the computation costs of LSGDR and gKDR seem not to be so expensive. Since the models (d) and (e) include variables from non-Gaussian densities such as a gamma and truncated normal density, Table 1 (d) and (e) imply that LSGDR works well for data drawn from a wide range of densities.

4.2. Regression Performance on Benchmark Datasets

Finally, we apply LSGDR, dMAVE, LSDR and gKDR to the UCI benchmark datasets (Bache and Lichman, 2013) and StatLib,[9] and investigate the regression performance of these methods. We randomly selected n samples from each dataset in the training phase, and the rest, whose number is denoted by n_{te}, were used in the test phase. After estimating $\mathbf{B}$ by each method in the training phase, we learned a regressor $f_{\mathrm{LR}}(\widehat{\mathbf{B}}^\top \boldsymbol{x})$ by ℓ_2-regularized least-squares with Gaussian kernels. In the test phase, the regression error was measured by $\sqrt{\frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \{y_i - f_{\mathrm{LR}}(\widehat{\mathbf{B}}^\top \boldsymbol{x}_i)\}^2}$. To make the estimation more challenging, we added more noise dimensions, which are independently drawn from $\mathrm{Gamma}(1,2)$, to the original data. In this experiment, since the true intrinsic dimensionality $d_{\mathbf{z}}$ is unknown, we chose it by cross-validation: Each method performed 5-fold cross-validation to choose the intrinsic dimensionality $d_{\mathbf{z}}$ from the candidates $\{1, 2, \ldots, d_{\mathbf{x}} - 1\}$ such that the regression error is minimized. Unlike the last experiments, we did not perform the comparison in terms of CPU time because LSGDR and gKDR provide computationally more efficient procedures than dMAVE and LSDR: The eigenvalue decomposition allows us to check all $\widehat{\mathbf{B}}$ to the candidates for $d_{\mathbf{z}}$ by estimating the gradients only once. On the other hand, since dMAVE and LSDR are not based on the eigenvalue decomposition, the same idea cannot be applied and thus they are computationally very expensive in this cross-validation.

When no noise dimension is added, LSGDR produces the best result or is comparable to the best method for most of the datasets, while for Housing and Power plant, other methods or the original data without dimension reduction provide better performance than LSGDR (Table 2). However, as the noise dimensions are added, LSGDR tends to significantly outperform the other methods. These results indicate that LSGDR is better at finding informative subspaces from high-dimensional data than other methods.

9. http://lib.stat.cmu.edu/datasets/

Table 2: The means and standard deviations of regression errors over 30 runs for each dataset. The best method in terms of the mean error and comparable methods according to the Wilcoxon signed-rank test at the significance level 1% are specified by bold face. d_{add} denotes the number of added noise dimensions to the original data.

White wine ($d_{\mathbf{x}} = 11 + d_{\text{add}}, n = 500$)

	LSGDR	dMAVE	LSDR	gKDR	No reduc.
$d_{\text{add}} = 0$	**0.839(0.012)**	0.850(0.015)	0.850(0.012)	0.845(0.014)	**0.838(0.009)**
$d_{\text{add}} = 2$	**0.844(0.010)**	0.853(0.012)	0.855(0.015)	0.852(0.011)	**0.846(0.010)**
$d_{\text{add}} = 4$	**0.848(0.011)**	0.862(0.017)	0.865(0.012)	0.860(0.021)	**0.851(0.009)**
$d_{\text{add}} = 6$	**0.849(0.010)**	0.868(0.017)	0.879(0.017)	0.864(0.020)	0.858(0.014)

Red wine ($d_{\mathbf{x}} = 11 + d_{\text{add}}, n = 500$)

	LSGDR	dMAVE	LSDR	gKDR	No reduc.
$d_{\text{add}} = 0$	**0.805(0.015)**	**0.807(0.018)**	**0.806(0.020)**	**0.804(0.016)**	**0.801(0.015)**
$d_{\text{add}} = 2$	**0.812(0.018)**	**0.817(0.020)**	**0.815(0.015)**	**0.814(0.014)**	**0.811(0.015)**
$d_{\text{add}} = 4$	**0.813(0.015)**	0.823(0.014)	**0.819(0.017)**	0.822(0.014)	0.821(0.012)
$d_{\text{add}} = 6$	**0.813(0.014)**	0.828(0.012)	0.827(0.013)	0.828(0.019)	0.826(0.012)

Housing ($d_{\mathbf{x}} = 13 + d_{\text{add}}, n = 200$)

	LSGDR	dMAVE	LSDR	gKDR	No reduc.
$d_{\text{add}} = 0$	0.464(0.045)	**0.438(0.038)**	0.472(0.054)	**0.433(0.043)**	**0.450(0.048)**
$d_{\text{add}} = 2$	**0.471(0.042)**	**0.465(0.041)**	**0.481(0.050)**	**0.463(0.040)**	**0.470(0.050)**
$d_{\text{add}} = 4$	**0.466(0.043)**	**0.466(0.042)**	0.489(0.042)	**0.459(0.046)**	**0.474(0.047)**
$d_{\text{add}} = 6$	**0.468(0.041)**	0.494(0.038)	0.513(0.060)	0.490(0.035)	0.526(0.044)

Concrete ($d_{\mathbf{x}} = 8 + d_{\text{add}}, n = 500$)

	LSGDR	dMAVE	LSDR	gKDR	No reduc.
$d_{\text{add}} = 0$	**0.416(0.019)**	0.420(0.021)	0.441(0.024)	**0.409(0.019)**	0.428(0.014)
$d_{\text{add}} = 2$	**0.426(0.026)**	0.434(0.020)	0.449(0.020)	**0.421(0.020)**	0.468(0.015)
$d_{\text{add}} = 4$	**0.421(0.023)**	0.447(0.026)	0.455(0.021)	0.446(0.022)	0.507(0.018)
$d_{\text{add}} = 6$	**0.418(0.018)**	0.454(0.021)	0.457(0.018)	0.462(0.020)	0.545(0.019)

Power plant ($d_{\mathbf{x}} = 4 + d_{\text{add}}, n = 500$)

	LSGDR	dMAVE	LSDR	gKDR	No reduc.
$d_{\text{add}} = 0$	0.255(0.003)	**0.253(0.003)**	0.255(0.003)	**0.253(0.003)**	**0.252(0.002)**
$d_{\text{add}} = 2$	0.257(0.003)	**0.254(0.004)**	0.257(0.003)	**0.256(0.003)**	0.264(0.003)
$d_{\text{add}} = 4$	**0.257(0.003)**	**0.255(0.002)**	0.258(0.002)	**0.258(0.006)**	0.281(0.004)
$d_{\text{add}} = 6$	**0.258(0.004)**	**0.257(0.003)**	0.259(0.003)	0.260(0.004)	0.294(0.005)

Body fat (*StatLib) ($d_{\mathbf{x}} = 13 + d_{\text{add}}, n = 100$)

	LSGDR	dMAVE	LSDR	gKDR	No reduc.
$d_{\text{add}} = 0$	**0.588(0.038)**	**0.600(0.031)**	**0.603(0.049)**	**0.606(0.040)**	0.613(0.029)
$d_{\text{add}} = 2$	**0.589(0.034)**	**0.610(0.047)**	0.621(0.036)	**0.612(0.046)**	0.623(0.028)
$d_{\text{add}} = 4$	**0.596(0.034)**	0.627(0.050)	0.636(0.056)	0.626(0.044)	0.642(0.029)
$d_{\text{add}} = 6$	**0.606(0.035)**	0.664(0.057)	0.654(0.053)	0.643(0.045)	0.661(0.030)

5. Conclusion

In this paper, we proposed a novel method for sufficient dimension reduction (SDR). Our main contribution in this paper was to develop a novel estimator for the gradients of logarithmic conditional densities, which is the key ingredient in our SDR method. The proposed gradient estimator is computationally efficient because all the solutions are computed in a closed from. Furthermore, a cross-validation method was also provided to objectively determine all the tuning parameters included in the estimator. Applying the proposed gradient estimator allowed us to develop a computationally efficient SDR method based on the eigenvalue decomposition to the expectation of the outer products of gradient estimates. We demonstrated that the proposed SDR method outperforms existing SDR methods in terms of both estimation accuracy and computational efficiency on artificial datasets as well as of regression accuracy on benchmark datasets especially for high-dimensional data.

In this paper, the proposed gradient estimator has been applied only for SDR. However, we believe that it has a wide range of applications. In future, we will explore novel applications in statistical data analysis.

Acknowledgments

H. Sasaki is supported by KAKENHI 15H06103, V. Tangkaratt is supported by KAKENHI 23120004, and M. Sugiyama is supported by KAKENHI 25700022.

References

K.P. Adragni and R.D. Cook. Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4385–4405, 2009.

K. Bache and M. Lichman. UCI machine learning repository, 2013.

C.J.C. Burges. Dimension reduction: A guided tour. *Foundations and Trends® in Machine Learning*, 2(4):275–365, 2009.

M.Á. Carreira-Perpiñán. A review of dimension reduction techniques. *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, pages 1–69, 1997.

F. Chiaromonte and R. D. Cook. Sufficient dimension reduction and graphics in regression. *Annals of the Institute of Statistical Mathematics*, 54(4):768–795, 2002.

R. D. Cook. *Regression Graphics: Ideas for Studying Regressions Through Graphics*. John Wiley & Sons, 1998.

R.D. Cook. SAVE: a method for dimension reduction and graphics in regression. *Communications in Statistics-Theory and Methods*, 29(9-10):2109–2121, 2000.

D. D. Cox. A penalty method for nonparametric estimation of the logarithmic derivative of a density function. *Annals of the Institute of Statistical Mathematics*, 37(1):271–288, 1985.

J. Fan and I. Gijbels. *Local Polynomial Modelling and its Applications*, volume 66. CRC Press, 1996.

K. Fukumizu and C. Leng. Gradient-based kernel dimension reduction for regression. *Journal of the American Statistical Association*, 109(505):359–370, 2014.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

M. Hristache, A. Juditsky, J. Polzehl, and V. Spokoiny. Structure adaptive approach for dimension reduction. *The Annals of Statistics*, 29(6):1537–1566, 2001.

A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

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

K.C. 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.

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

A. M. Samarov. Exploring regression structure using nonparametric functional estimation. *Journal of the American Statistical Association*, 88(423):836–847, 1993.

H. Sasaki, A. Hyvärinen, and M. Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. In *Machine Learning and Knowledge Discovery in Databases Part III- European Conference, ECML/PKDD 2014*, volume 8726, pages 19–34, 2014.

B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. CRC press, 1986.

T. Suzuki and M. Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. *Neural Computation*, 25(3):725–758, 2013.

V. Tangkaratt, N. Xie, and M. Sugiyama. Conditional density estimation with dimensionality reduction via squared-loss conditional entropy minimization. *Neural computation*, 27(1):228–254, 2015.

Y. Xia. A constructive approach to the estimation of dimension reduction directions. *The Annals of Statistics*, 35(6):2654–2690, 2007.

Y. Xia, H. Tong, W. K. Li, and L. X. Zhu. An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):363–410, 2002.