Nonlinear Dimension Reduction with Kernel Sliced Inverse Regression

Yi-Ren Yeh[1], Su-Yun Huang[2] and Yuh-Jye Lee[1*]

[1]Computer Science and Information Engineering
National Taiwan University of Science and Technology

[2]Institute of Statistical Science, Academia Sinica

October 9, 2007

Abstract

Sliced inverse regression (SIR) is a renowned dimension reduction method for finding an effective low-dimensional linear subspace. Like many other linear methods, SIR can be extended to nonlinear setting via the "kernel trick". The main focus of this article is twofold. The first is on the implementation algorithm of kernel SIR for fast computation. The other is on kernel SIR's ability to combine with other linear learning algorithms for classification and regression. Numerical experiments show that kernel SIR is an effective kernel tool for nonlinear dimension reduction and it can easily combine with other linear algorithms to form a powerful toolkit for nonlinear data analysis.

Keywords: dimension reduction, eigenvalue decomposition, kernel, singular value decomposition, sliced inverse regression, support vector machines.

*No. 43, Sec. 4, Keelung Road, Taipei, TAIWAN

Tel.:+886 2 27301066; Fax:+886 2 27301081

1 Introduction

Dimension reduction is an important topic in machine learning and data mining. The main demand comes from complex data analysis, data visualization and parsimonious modeling (Alpaydm, 2004; Cook, 1998). Modern data are usually complex, high-dimensional and with nonlinear structures. A dimension reduction technique helps us to characterize the key data structure using only a few main features ranked by their importance. It thus provides a way for data visualization to gain better intuitive insights of the underlying data. The most popular dimension reduction method is probably the principal component analysis (PCA), which is an unsupervised method. In the contrast, the sliced inverse regression (SIR) (Li, 1991) extracts the dimension reduction subspace based on the covariance matrix of input attributes inversely regressed on the responses. SIR can be viewed as a supervised companion of PCA for linear dimension reduction. SIR has won its reputation to perform well in dimension reduction and related applications and has gained great attention in statistical literature (Chen & Li, 1998; Cook, 1998; Duan & Li, 1991; Hall & Li, 1993; Li, 1991, 1997). The work by Wu (Wu, n.d.) extends the classical SIR to nonlinear dimension reduction via the kernel method. This extension is named kernel sliced inverse regression (KSIR), and is applied to support vector classification. In this article we go for a further study and emphasize on KSIR's implementation technique, its ability to combine with other linear algorithms and its applications to support vector classification as well as regression.

The subsequent sections are organized as follows. Section 2 gives a brief review of the classical SIR. Section 3 introduces its kernel extension in a reproducing kernel Hilbert space (RKHS) framework and provides some insight into the technical conditions. Theory that leads to the estimation of feature dimension reduction subspace is given. In the same section, a fast implementation algorithm is prescribed and some numerical issues are discussed. Section 4 is on numerical experiments and results. Concluding remarks are in Section 5. Some further theoretical properties for KSIR are in the Appendix.

2 Sliced Inverse Regression

Different from other dimension reduction methods, SIR summarizes a regression or classification model as follows:

$$\mathbf{y} = f(\beta_1'\mathbf{x}, \ldots, \beta_d'\mathbf{x}; \epsilon), \quad \beta_k, \mathbf{x} \in \mathbb{R}^p, \tag{1}$$

where d (often $\ll p$) is the effective dimensionality and $\{\beta_1, \ldots, \beta_d\}$ forms a basis of the effective dimension reduction (*e.d.r.*) subspace. The model above implies that most of the relevant information in $\mathbf{x}$ about $\mathbf{y}$ is contained in $\{\beta_1'\mathbf{x}, \ldots, \beta_d'\mathbf{x}\}$. The dimensionality of input attributes gets cut down from p to d. The model (1) does not impose any structure on f, which can be any linear or nonlinear form. This model has only assumed that the effects of input attributes $\mathbf{x}$ on the output variable $\mathbf{y}$ can be characterized by a certain low-dimensional projection onto the linear subspace spanned by $\{\beta_1, \ldots, \beta_d\}$. That is to say, the reduced input attributes $(\beta_1'\mathbf{x}, \ldots, \beta_d'\mathbf{x})$ carry as much information about $\mathbf{y}$ as the original $\mathbf{x}$. The model (1) takes the weakest form for linear dimension reduction. It only assumes the existence of some low-dimensional linear subspace without imposing any parametric structure on f. With this model (1) and the linear design condition (LDC) defined below, SIR then extracts the *e.d.r.* subspace by using the notion of inverse regression (Duan & Li, 1991; Li, 1991). Define the central inverse regression function as follows:

$$\mathbf{g}(\mathbf{y}) = E(\mathbf{x}|\mathbf{y}) - E(\mathbf{x}) \in \mathbb{R}^p.$$

We say that $\{\beta_1, \ldots, \beta_d\}$ satisfies the LDC, if, for any $b \in \mathbb{R}^p$, the conditional expectation $E(b'\mathbf{x}|\beta_1'\mathbf{x}, \ldots, \beta_d'\mathbf{x})$ is affine linear in $\{\beta_1'\mathbf{x}, \ldots, \beta_d'\mathbf{x}\}$. That is, there exist some constants $c_0, c_1, \ldots, c_d$ such that

$$E(b'\mathbf{x}|(\beta_1'\mathbf{x}, \ldots, \beta_d'\mathbf{x})) = c_0 + c_1\beta_1'\mathbf{x} + \cdots + c_d\beta_d'\mathbf{x}. \tag{2}$$

With the model (1) and the LDC (2), it can be shown (Li, 1991) that the basis of the *e.d.r.* subspace can be estimated by the leading directions from the generalized eigenvalue

decomposition of $Cov(\mathbf{g})$, denoted by $\Sigma_{E(\mathbf{x}|\mathbf{y})}$, with respect to $Cov(\mathbf{x})$, denoted by $\Sigma_{\mathbf{x}}$. SIR is devised to find such leading directions. In other words, with $\mathbf{x}$ being standardized, SIR finds the leading directions that the central inverse regression function $\mathbf{g}(\mathbf{y})$ has the largest variation. These are the most informative directions in the input space for describing $\mathbf{y}$. In practical supervised learning tasks, the joint distribution of the input vector $\mathbf{x}$ and the output variable $\mathbf{y}$ is unknown but fixed. We use bold-faced $\mathbf{x}$ and $\mathbf{y}$ for random vectors and variables and italic letters x and y for their realizations. Suppose we have a data set

$$\mathcal{D} := \{(x^1, y_1), \ldots, (x^n, y_n)\},$$

each pair (x^i, y_i) is an instance $x^i \in \mathbb{R}^p$ with its response or class label y_i. Let $A \in \mathbb{R}^{n \times p}$ be the data matrix of input attributes and $Y = (y_1, \ldots, y_n)' \in \mathbb{R}^n$ be the corresponding responses. Each row of A represents an observation, x^i. The empirical data version of sliced inverse regression finds the dimension reduction directions by solving the following generalized eigenvalue problem based on empirical data $\mathcal{D}$:

$$\Sigma_{E(A|Y_J)}\beta = \lambda \Sigma_A \beta, \tag{3}$$

where Σ_A is the sample covariance matrix of A, Y_J denotes the membership of slices and there are J many slices, and $\Sigma_{E(A|Y_J)}$ denotes the between-slice sample covariance matrix based on sliced means given by

$$\Sigma_{E(A|Y_J)} = \frac{1}{n} \sum_{j=1}^{J} n_j (\bar{x}^j - \bar{x})(\bar{x}^j - \bar{x})'.$$

Here $\bar{x}$ is the sample grand mean, $\bar{x}^j = \frac{1}{n_j} \sum_{i \in S_j} x^i$ is the sample mean for the jth slice and S_j is the index set for jth slice. Note that the slices are extracted from A according to the sorted responses Y. For classification, $\bar{x}^j$ is simply the sample mean of input attributes for the jth class. There is an equivalent way to modeling SIR. We consider the following optimization problem

$$\max_{\beta \in \mathbb{R}^p} \ \beta' \Sigma_{E(A|Y_J)} \beta \ \text{ subject to } \beta' \Sigma_A \beta = 1. \tag{4}$$

The solution, denoted by β_1, gives the first *e.d.r.* direction such that class means projected along β_1 are most spreading out, where β_1 is normalized with respect to the sample covariance matrix Σ_A. Repeatedly solving this optimization problem with the orthogonality constraints $\beta_k \Sigma_A \beta_l = \delta_{k,l}$, where $\delta_{k,l}$ is the Kronecker delta, the sequence of solutions $\beta_1, \ldots, \beta_d$ form the *e.d.r.* basis. Some insightful discussion to enhance the SIR methodology and applications can be found in Chen and Li (Chen & Li, 1998).

3 Kernel Extension for SIR

The classical SIR is designed to find a *linear* transformation from the input space to a low dimensional *e.d.r.* subspace that keeps as much information as possible for the output variable $\mathbf{y}$. However, it does not work for nonlinear feature extraction and it fails to find linear directions being in the null space or having small angles to the null space of $\Sigma_{E(\mathbf{x}|\mathbf{y})}$. The following regression example taken from Friedman (Friedman, 1991) is used for illustrative purpose for KSIR. This example has explanatory variables in $\mathbb{R}^{10}$:

$$\mathbf{y} = f(\mathbf{x}_1, \ldots, \mathbf{x}_{10}; \epsilon) = 10 \sin(\pi \mathbf{x}_1 \mathbf{x}_2) + 20(\mathbf{x}_3 - 0.5)^2 + 10\mathbf{x}_4 + 5\mathbf{x}_5 + \epsilon, \qquad (5)$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_{10})$ are independent and identically distributed (iid) uniform random variables over $[0, 1]$ and $\epsilon \overset{iid}{\sim} N(0, 1)$. SIR fails to find the direction along the x_3-coordinate due to its symmetric structure to the vertical axis at $x_3 = 0.5$. The variance of $E(\mathbf{x}_3|\mathbf{y})$ is zero and hence scaled principal components based on $\Sigma_{E(\mathbf{x}|\mathbf{y})}$ will not find the direction of x_3-axis. For the Friedman example, the function's key features can be easily described by a few nonlinear components extracted by KSIR. Experimental study of it can be found in a later section.

3.1 Geometric framework and properties

In SIR, the model assumption (1) says that there exists a linear dimension reduction subspace and that the underlying objective function f can be any linear or nonlinear

form in that subspace. In looking for a nonlinear extension, the original input space is embedded to a high dimensional feature space $\mathcal{Z}$ via the feature map

$$\Phi : \mathcal{X} \subset \mathbb{R}^p \mapsto \mathcal{Z}, \tag{6}$$

where $\Phi(x)$ is the kernel spectrum for a certain positive definite kernel, i.e., $K(x,u) = \Phi(x)'\Phi(u)$. The following dimension reduction model in the feature space $\mathcal{Z}$ is assumed

$$\mathbf{y} = f(\beta_1'\mathbf{z}, \ldots, \beta_d'\mathbf{z}; \epsilon), \quad \beta_k, \mathbf{z} \in \mathcal{Z}. \tag{7}$$

In other words, there exist $\beta_1, \ldots, \beta_d \in \mathcal{Z}$ such that $\mathbf{y}$ and $\mathbf{z}$ are conditionally independent given $\{\beta_1'\mathbf{z}, \ldots, \beta_d'\mathbf{z}\}$. See Wu (Wu, n.d.) for the kernel extension of SIR in the framework of $\mathcal{Z}$. As the feature space $\mathcal{Z}$ is often not explicitly known to us, we will then look for a substitute with explicit expression. As part of the key purposes of dimension reduction are feature extraction and data visualization, a concrete feature space is necessary. We, therefore, transform the feature space $\mathcal{Z}$ to an isometric isomorphic space, which is explicitly known and where data can be observed. Consider an alternative feature map $\Gamma : \mathcal{X} \mapsto \mathcal{H}_K$ given by

$$x \mapsto \Gamma(x) := K(x, \cdot). \tag{8}$$

Kernels used here are positive definite, also known as reproducing kernels. For a given positive definite kernel K, its associated Hilbert space consists of all finite kernel mixtures $\sum_{i=1}^m a_i K(x, u_i)$ and their limits, where $m \in \mathbb{N}$, $u_i \in \mathbb{R}^p$ and $a_i \in \mathbb{R}$ all can be arbitrary. This Hilbert space is known as the reproducing kernel Hilbert space (RKHS), denoted by $\mathcal{H}_K$. Throughout this article we assume that all the reproducing kernels employed are (C1) symmetric (i.e., $K(x,u) = K(u,x)$) and measurable, (C2) of trace type, i.e., $\int_{\mathcal{X}} K(x,x)d\mu < \infty$ and (C3) for $x \neq u$, $K(x,\cdot) \neq K(u,\cdot)$ in $L_2(\mathcal{X}, \mu)$ sense for some underlying continuous probability distribution μ. The distribution μ need not be the same as the distribution of $\mathbf{x}$. The original input space $\mathcal{X}$ is then embedded into a new feature space $\mathcal{H}_K$ via the transformation Γ. Each input point $x \in \mathcal{X}$ is mapped to an element $K(x,\cdot) \in \mathcal{H}_K$. Let $\mathcal{J} : \mathcal{Z} \mapsto \mathcal{H}_K$ be a map from the spectrum-based feature

space $\mathcal{Z}$ to the kernel associated Hilbert space $\mathcal{H}_K$ defined by $\mathcal{J}(\Phi(x)) = K(x, \cdot)$. By condition (C2) and the reproducing property

$$\langle K(x, \cdot), f(\cdot)\rangle_{\mathcal{H}_K} = f(x), \quad \forall f \in \mathcal{H}_K, \forall x \in \mathcal{X},$$

it is easy to verify that $\mathcal{J}$ is a one-to-one linear transformation satisfying

$$\|z\|_{\mathcal{Z}}^2 = \|\Phi(x)\|_{\mathcal{Z}}^2 = K(x, x) = \|K(x, \cdot)\|_{\mathcal{H}_K}^2 = \|\mathcal{J}(z)\|_2^2.$$

Thus, $\Phi(\mathcal{X})$ and $\Gamma(\mathcal{X})$ are isometrically isomorphic, and the two feature representations (6) and (8) are equivalent in this sense. We will work directly on the latter feature representation (8), which is explicit and concrete.

The classical SIR solves a generalized spectrum decomposition of the between-slice covariance matrix in the pattern Euclidean space $\mathbb{R}^p$. Similarly, KSIR solves a generalized spectrum decomposition of the between-slice covariance operator in the feature RKHS $\mathcal{H}_K$. The following two definitions place the notions of *e.d.r.* subspace and LDC in the framework of $\mathcal{H}_K$.

Definition 1 (e.d.r. subspace in $\mathcal{H}_K$) *Let $H = \{h_1, \ldots, h_d\}$ be a collection of elements in $\mathcal{H}_K$, and let $\mathcal{H}$ be the linear subspace spanned by elements in H. We say that $\mathcal{H}$ is an e.d.r. subspace if $\mathbf{y}$ and $\mathbf{x}$ are conditionally independent given $\{h_1(\mathbf{x}), \ldots, h_d(\mathbf{x})\}$, i.e., information about $\mathbf{y}$ in $\mathbf{x}$ is contained in $\{h_1(\mathbf{x}), \ldots, h_d(\mathbf{x})\}$. We name h_k's the e.d.r. directions, $\mathcal{H}$ the e.d.r. subspace and $h_k(\mathbf{x})$'s the e.d.r. variates.*

One can picture $h_k \in \mathcal{H}_K$ as the image of $\beta_k \in \mathcal{Z}$ via $\mathcal{J}$ and β_k as the pre-image of h_k. This gives an interplay between the *e.d.r.* directions in $\mathcal{Z}$ and in $\mathcal{H}_K$. Note that, by the isomorphism and the kernel reproducing property, we have

$$\beta_k' \mathbf{z} = \langle h_k(\cdot), K(\mathbf{x}, \cdot)\rangle_{\mathcal{H}_K} = h_k(\mathbf{x}).$$

Then the *e.d.r.* model (7) becomes

$$\mathbf{y} = f(h_1(\mathbf{x}), \cdots, h_d(\mathbf{x}); \epsilon),$$

where $h_1(\mathbf{x}), \ldots, h_d(\mathbf{x})$ are nonlinear functional variates of $\mathbf{x}$. Let $\ell : \mathcal{H}_K \mapsto \mathbb{R}$ be an arbitrary linear functional. By Riesz representation Theorem (Wahba, 1999), there exists an $f \in \mathcal{H}_K$ so that

$$\ell(K(\mathbf{x}, \cdot)) = \langle K(\mathbf{x}, \cdot), f(\cdot) \rangle_{\mathcal{H}_K} = f(\mathbf{x}). \tag{9}$$

Conversely, any $f \in \mathcal{H}_K$ defines a linear functional through (9). Thus, the functional variate $f(\mathbf{x})$ can be viewed as an analog to $b'\mathbf{x}$. It leads to the functional version of the linear design condition for KSIR.

Definition 2 (linear design condition in $\mathcal{H}_K$) *Let $H = \{h_1, \ldots, h_d\}$ be a collection of elements in $\mathcal{H}_K$. H is say to satisfy the linear design condition, if the following statement holds. For any $f \in \mathcal{H}_K$,*

$$E(f(\mathbf{x})|h_1(\mathbf{x}), \ldots, h_d(\mathbf{x})) = c_0 + c_1 h_1(\mathbf{x}) + \cdots + c_d h_d(\mathbf{x}) \tag{10}$$

for some constants $c_0, c_1, \ldots, c_d$.

At the first glance, this LDC seems more restrictive than the one given by (2) for the classical SIR. Actually, it is not the case. Condition (10) says that the regression of f on h_k is affine linear where h_k's are kernel mixtures yet to be estimated. As kernel functions are abundant and flexible building blocks, it can be shown that any smooth function can be well approximated by a kernel mixture (Saitoh, 1997; Thompson & Tapia, 1990). Therefore, condition (10) is not as restrictive as the LDC (2) in the Euclidean space. The philosophy is that, linearity in an Euclidean space is much stricter than linearity in an RKHS. The former is in terms of linear combinations of $x_1, \ldots, x_p$, while the latter is in terms of linear combinations of kernels, which form a large body of functions of various shapes. In fact, the linearity in $\mathcal{H}_K$ is equivalent to the linearity in the feature space $\mathcal{Z}$. An elliptically symmetric distribution of data scatter in the *e.d.r.* subspace will ensure the validity of LDC. Some mild departure from the elliptical symmetry in the *e.d.r.* subspace will not hurt the application of SIR (Li, 1991, 1997). For KSIR, the kernel transform has mapped the data into a very high dimensional feature Hilbert

space, and we look for low-dimensional projections therein. Low-dimensional projections from high-dimensional data are known to be able to improve the elliptical symmetry of data distribution (Diaconis & Freedman, 1984; Hall & Li, 1993).

The goal of KSIR is to estimate the *e.d.r.* directions h_k's. Note that the mapped image $K(\mathbf{x}, \cdot)$ is a random element in $\mathcal{H}_K$. Similar to the classical SIR, KSIR finds the *e.d.r.* directions by solving a generalized spectrum decomposition problem. In practice and in finite sample case, we have to work on an approximation subspace of $\mathcal{H}_K$ with a finite basis set, say $\{K(\cdot, A)\} = \{K(\cdot, x^1), \ldots, K(\cdot, x^n)\}$. Other choices of basis sets are fine, too, as long as they represent a distribution similar to the distribution of training input attributes. It is the key idea of our approximation to KSIR in Section 3.2. For simplicity and better intuition, we will work on a finite basis approximation subspace in the main body of this article. The functional version is given in the Appendix for interested readers. With the finite basis set approximation, the *e.d.r.* variates and an arbitrary nonlinear variate $f(\mathbf{x})$ can be represented as

$$h_k(\mathbf{x}) = K(\mathbf{x}, A)\alpha_k \text{ for some } \alpha_k \in \mathbb{R}^n, \quad k = 1, \ldots, d, \quad \text{and}$$

$$f(\mathbf{x}) = K(\mathbf{x}, A)a \text{ for some } a = (a_1, \ldots, a_n) \in \mathbb{R}^n.$$

Let $T = K(\mathbf{x}, A)'$ a random column vector in $\mathbb{R}^n$. The LDC in Definition 2 can be restated as shown below using finite basis approximation:

$$E(a'T|\alpha_1'T, \ldots, \alpha_d'T) = c_0 + c_1\alpha_1'T + \cdots + c_d\alpha_d'T, \quad \forall a \in \mathbb{R}^n. \tag{11}$$

With this finite basis, kernel data are given by $\{K(x^i, A)\}_{i=1}^n$. Collect them into a matrix form (observations by row and variables by column) we get the kernel data matrix $K = K(A, A) \in \mathbb{R}^{n \times n}$ where $K(A, A)_{ij} = \Phi(x^i)'\Phi(x^j)$. The second argument A in $K(A, A)$ is used for kernel basis. With the introduction of a finite basis set, the *e.d.r.* directions and the subspace can be described in terms of finite-dimensional vectors and vector spaces. By taking such basis set $K(\cdot, A)$, the KSIR procedure is simply the classical SIR with $T = K(\mathbf{x}, A)'$ and $\mathbf{y}$. Hence Theorem 3.1 of Li (Li, 1991) leads to the following theorem.

Theorem 1 *Assume the existence of an e.d.r. subspace* $\mathcal{H} = \text{span}\{K(\mathbf{x}, A)\alpha_1, \ldots, K(\mathbf{x}, A)\alpha_d\}$ *and the LDC (11). Then the central inverse regression vector falls into the subspace spanned by* $\{\Sigma_T\alpha_1, \ldots, \Sigma_T\alpha_d\}$*, i.e.,*

$$E(T|\mathbf{y}) - E(T) \in \text{span}\{\Sigma_T\alpha_1, \ldots, \Sigma_T\alpha_d\}, \tag{12}$$

where Σ_T *is the covariance matrix of* $T = K(\mathbf{x}, A)'$*.*

Theorem 1 is stated in terms of a finite-dimensional approximation using the basis set $K(\cdot, A)$ for practical simplicity. A more general functional version of the theorem can be found in the Appendix. It formulates KSIR as a generalized spectrum decomposition of the between-slice covariance operator in an RKHS framework. From (12) we see that the central inverse regression of T on $\mathbf{y}$ degenerates at any directions orthogonal to $\text{span}\{\Sigma_T\alpha_1, \ldots, \Sigma_T\alpha_d\}$. Thus, the covariance matrix of $E(T|\mathbf{y}) - E(T)$ provides a way for estimating the *e.d.r.* subspace $\mathcal{H}$. In other words, to estimate the *e.d.r.* directions, we solve the following generalized eigenvalue problem:

$$\Sigma_{E(T|\mathbf{y})}\alpha = \lambda\Sigma_T\alpha \tag{13}$$

or equivalently

$$\max_{\alpha\in\mathbb{R}^p} \alpha'\Sigma_{E(T|\mathbf{y})}\alpha \text{ subject to } \alpha'\Sigma_T\alpha = 1. \tag{14}$$

Equation (13) is only used for illustrative purpose. We will not solve this generalized eigenvalue problem directly. Instead, we implement KSIR in a different way for numerical consideration and fast computation. The implementation is explained below. Let π_j be the proportional size (or prior probability) of the jth slice. Consider the eigenvalue decomposition of $W'\Sigma_T^{-1}W$ as UDU', where W consists of centered and weighted slice means. Precisely, the jth column of W is given by

$$w_j = \sqrt{\pi_j}\left(E(T|\mathbf{y}_i, i \in S_j) - E(T)\right).$$

Since the eigenvector(s) associated with the zero eigenvalue has no information for y, we may restrict U and D to eigenvectors with non-zero eigenvalues. For simplicity, assume

there is only one zero eigenvalue. Then, U has size $J \times (J-1)$ and whose columns are formed by eigenvectors and D is a $(J-1) \times (J-1)$ diagonal matrix with descending and non-zero eigenvalues. From Theorem 1, we know that the column space spanned by $\Sigma_T^{-1}W$, denoted by $\mathcal{C}(\Sigma_T^{-1}W)$, provides a way of estimating the *e.d.r.* subspace. However, it only tells that $\mathcal{C}(\Sigma_T^{-1}W)$ is part (or the whole) of the *e.d.r.* subspace, but it does not provide the individual orthonormal basis vectors. In the following Proposition, we will give the orthonormal basis set, which are simply the *e.d.r.* directions.

Proposition 2 (e.d.r. directions) *The orthonormalized e.d.r. directions are given by columns of* $\Sigma_T^{-1}WUD^{-1/2}$.

Proof: Since $\mathcal{C}(\Sigma_T^{-1}W)$ is in the *e.d.r.* subspace, we are then looking for an orthonormal basis, denoted by V, for $\mathcal{C}(\Sigma_T^{-1}W)$, where the orthonormality is in terms of $V'\Sigma_T V = I$. The singular value decomposition (SVD) is the most direct way to find orthonormal basis for column space for a given matrix. As the normalization is in terms of $V'\Sigma_T V = I$, a common SVD is applied to the matrix $\Sigma_T^{-1/2}W$. Since only right singular vectors are needed for column orthonormalization, the right singular vectors can be solved from the eigenvalue decomposition of the following square matrix:

$$(\Sigma_T^{-1/2}W)'(\Sigma_T^{-1/2}W) = W'\Sigma_T^{-1}W = UDU',$$

where D is a diagonal matrix of size $(J-1)\times(J-1)$ with descending nonzero eigenvalues and U is a matrix of size $J \times (J-1)$ consisting of associated eigenvectors. Let $V = \Sigma_T^{-1}WUD^{-1/2}$. Its columns are still in the column space $\mathcal{C}(\Sigma_T^{-1}W)$ and hence are still in the *e.d.r.* subspace. It is then only left to check the orthonormality:

$$V'\Sigma_T V = (D^{-1/2}U'W'\Sigma_T^{-1})\Sigma_T(\Sigma_T^{-1}WUD^{-1/2}) = D^{-1/2}U'UDU'UD^{-1/2} = I.$$

The proof is completed. □

Note that if only a few leading directions are needed, we can use only leading columns from U and corresponding leading diagonal elements from D.

Proposition 2 is a population version. In practical data analysis sample estimates based on the given data $\mathcal{D}$ are used to replace all the population-based quantities. The sample covariance matrices $\Sigma_{E(K|Y_J)}$ and Σ_K are used to replace the population covariance matrices $\Sigma_{E(T|\mathbf{y})}$ and Σ_T, respectively. Also the following sample estimates for the centered weighted slice means are used to replace their population versions:

$$w_j = \sqrt{n_j/n}\left(\frac{\mathbf{1}'_{n_j}K(A_{S_j},A)}{n_j} - \frac{\mathbf{1}'_n K(A,A)}{n}\right)',$$

where $\mathbf{1}'_{n_j}K(A_{S_j},A)/n_j$ and $\mathbf{1}'_n K(A,A)/n$ are respectively the jth slice sample mean and the grand mean of $K(A,A)$. Note that then $WW' = \Sigma_{E(K|Y_J)}$ is the between-slice sample covariance. The sample covariance matrix Σ_K is singular and is often having much lower effective rank than its size. This low-effective-rank phenomenon causes numerical instability and poor *e.d.r.* directions estimation. We will discuss this issue and its remedy in next subsection.

3.2 Approximation to KSIR for fast computation

In many real world applications, the effective rank of the covariance matrix of kernel data is very low. This causes the numerical instability and leads to inferior estimation of the *e.d.r.* directions. Adding a ridge-type regularization term is a common way to solve the numerical instability. That is to add a small diagonal matrix εI to Σ_K. This ridge-type regularization though lessens the numerical instability, it does not solve the inferior estimation problem. The kernel data matrix has much lower effective rank than the data size n. Ridge-type regularization acts like appending unnecessary and nuisance coordinate axes to the effective and useful axes. Though the magnitude along each nuisance coordinate is small but there are many of them, which can add up to have an influential effect and leads to poor estimates. An appropriate way to deal with the problem is to find a reduced-column approximation to K, denoted by $\tilde{K}$, so that $\tilde{K}$ has full column rank and its column space $\mathcal{C}(\tilde{K})$ provides a good approximation to $\mathcal{C}(K)$. This approximation will enhance the numerical stability without much information loss.

Therefore, throughout this article we will adopt a reduced-column approximation instead of a ridge-type regularization for the singularity problem. The reduced-column approximation will cut down the problem size of the generalized spectrum decomposition required in the *e.d.r.* directions estimation and will also speed up the computation of it.

Let $\tilde{P}$ be a projection matrix of size $n \times \tilde{n}$, which satisfies $\tilde{P}'\tilde{P} = I_{\tilde{n}}$. Given a reduced-column kernel data $\tilde{K} := K\tilde{P}$, the approximation of KSIR is to solve the following reduced generalized eigenvalue problem:

$$\Sigma_{E(\tilde{K}|Y_J)}\tilde{\alpha} = \lambda\Sigma_{\tilde{K}}\tilde{\alpha}, \tag{15}$$

which is of much smaller size, as $\tilde{n} \ll n$. With the use of reduced kernel $\tilde{K}$, the corresponding centered weighted slice means are given by $\tilde{W}_{\tilde{n}\times J}$ with the jth column

$$\tilde{w}_j = \sqrt{n_j/n}\left(\frac{\mathbf{1}_{n_j}'\tilde{K}_{S_j}}{n_j} - \frac{\mathbf{1}_n'\tilde{K}}{n}\right)',$$

where $\mathbf{1}_{n_j}'\tilde{K}_{S_j}/n_j$ and $\mathbf{1}_n'\tilde{K}/n$ are respectively the jth slice mean and the grand mean of $\tilde{K}$. We can also apply Proposition 2 to the reduced problem (15) and the resulting *e.d.r.* directions are given by $\tilde{V} = \Sigma_{\tilde{K}}^{-1}\tilde{W}\tilde{U}\tilde{D}^{-1/2}$, where $\tilde{U}$ and $\tilde{D}$ are the eigenvectors and eigenvalues for $\tilde{W}'\Sigma_{\tilde{K}}^{-1}\tilde{W}$. Here, $\Sigma_{\tilde{K}}^{-1}$ exists if a proper projection $\tilde{P}$ is used. The KSIR algorithm using a reduced kernel approximation is given in Table 1. Two strategies for choosing a low-rank projection matrix $\tilde{P}$ are discussed after the algorithm.

Reduced kernel approximation by optimal basis. The SVD gives the optimal low-rank projection to get a reduced kernel. The SVD step aims to cut the number of kernel columns to its effective rank to avoid the numerical instability and to cope with the difficulty encountered in *e.d.r.* directions estimation. Consider the SVD of the centered full kernel matrix:

$$\left(I_n - \frac{\mathbf{1}_n\mathbf{1}_n'}{n}\right)K = G\Lambda P',$$

where $G'G = I$, $P'P = I$ and Λ is a diagonal matrix with descending singular values. Often the diagonal elements decay to zero very fast (Lee & Huang, 2007) and we need

Table 1: KSIR Algorithm

KSIR Algorithm
Input: reduced kernel matrix $\tilde{K}$ an $n \times \tilde{n}$ matrix and Y_J an n-vector.
Output: KSIR directions $V_{\tilde{n}\times(J-1)}$ and associated eigenvalues $d_{(J-1)\times 1}$.
1. Compute the centered and weighted slice means $\tilde{W}_{\tilde{n}\times J}$;
// J is the number of slices //
2. Compute the covariance matrix $\Sigma_{\tilde{K}}$ of the reduced kernel;
3. Compute the eigenvalue decomposition of $\tilde{W}'\Sigma_{\tilde{K}}^{-1}\tilde{W}$ as $\tilde{U}\tilde{D}\tilde{U}'$;
// $O(J^3)$ for solving the eigenvalue problem //
// $\tilde{D}$ and $\tilde{U}$ consist of non-zero eigenvalues and associated eigenvectors //
// $O(\tilde{n}^3)$ for solving the linear system $\Sigma_{\tilde{K}}X = \tilde{W}$ to get $\Sigma_{\tilde{K}}^{-1}\tilde{W}$ //
4. $V \leftarrow \Sigma_{\tilde{K}}^{-1}\tilde{W}\tilde{U}\tilde{D}^{-\frac{1}{2}}$; $d \leftarrow$ diagonal$\{\tilde{D}\}$.

only a small number of leading eigenvectors to approximate the centered K:

$$\left(I_n - \frac{\mathbf{1}_n\mathbf{1}_n'}{n}\right)K = G\Lambda P' \approx \tilde{G}\tilde{\Lambda}\tilde{P}',$$

where $\tilde{G}$ and $\tilde{P}$, both say of the size $n \times \tilde{n}$, consist of $\tilde{n}$ leading columns of G and P, respectively, and $\tilde{\Lambda}$, of size $\tilde{n} \times \tilde{n}$, consists of leading diagonals of Λ. We will work on the reduced-column kernel matrix $\tilde{K} := K\tilde{P}$ for the KSIR procedure. Note that Proposition 2 also applies to $\tilde{K}$, and P can be obtained from the eigenvalue decomposition of $Cov(K)$:

$$Cov(K) := \Sigma_K = PSP' \approx \tilde{P}\tilde{S}\tilde{P}', \text{ where } S = \Lambda^2 \text{ and } \tilde{S} = \tilde{\Lambda}^2.$$

Also note that

$$Cov(\tilde{K}) := \Sigma_{\tilde{K}} = \frac{1}{n}\tilde{P}'K\left(I_n - \frac{\mathbf{1}_n\mathbf{1}_n'}{n}\right)K\tilde{P} = \tilde{P}'\Sigma_K\tilde{P} = \tilde{S},$$

which makes the inverse of $\Sigma_{\tilde{K}}$ readily there. In other words, $\Sigma_{\tilde{K}}^{-1}$ in KSIR algorithm's Step 3 can be obtained from resulting matrices in the SVD preprocessing step. The

reduced-column matrix by leading singular vectors guarantees the linear independence among columns. However, this strategy only works for small to median sized kernel matrix, as for large kernel the computing cost for large eigenvalue problem is heavy and of complexity $O(n^3)$. Sometimes the large-scale data can go beyond the capacity of the memory size. This SVD step takes up most of the computing time in extracting KSIR directions. For massive data sets, it is not economic and sometimes can be computationally difficult or even impossible to compute the optimal basis from the full kernel. Thus, we provide another strategy to handle the large scale problem.

Reduced kernel approximation by random basis. For median to large sized data sets, we use the random subset reduced kernel to cut the kernel column size. In the random subset approach we choose $\tilde{P}$ as a column subset from I_n. In practice, it is not necessary to generate the full K nor to calculate the transformation $\tilde{K} := K\tilde{P}$. Instead we directly build up $\tilde{K}$ with selected columns only. The basic concept of random subset reduced kernel technique is to approximate the full kernel by the Nyström approximation:

$$K(A, A) \approx K(A, \tilde{A})K(\tilde{A}, \tilde{A})^{-1}K(\tilde{A}, A) = \tilde{K}K(\tilde{A}, \tilde{A})^{-1}\tilde{K}', \tag{16}$$

where $\tilde{A}_{\tilde{n}\times p}$ is a random subset of A and $K(A, \tilde{A}) = \tilde{K}_{n\times\tilde{n}}$ is a reduced kernel consisting of partial columns of the full kernel. Note that

$$K(A, A)\alpha \approx \tilde{K}K(\tilde{A}, \tilde{A})^{-1}\tilde{K}'\alpha = \tilde{K}\tilde{\alpha},$$

where $\tilde{\alpha} = K(\tilde{A}, \tilde{A})^{-1}\tilde{K}'\alpha$ is an approximation to the full problem. It means we only use $\tilde{n}$ basis functions $\{K(\cdot, \tilde{A})\}$ for modeling functions in $\mathcal{H}_K$. See (Lee & Huang, 2007) for more technical details and statistical properties for the random subset approach. The resulting reduced kernel matrix $\tilde{K}$ has full column rank so that $\Sigma_{\tilde{K}}$ is well-conditioned. The singularity problem can be resolved and the computational cost can be cut down at the same time. Our selection of reduced set is done by a stratified random sampling from the full set A. From results later in Section 4, we see that the reduced KSIR performs well while using only partial kernel columns (i.e., partial kernel basis). Note that the reduction ratio should be more degraded for larger data, since a small ratio of

a large data often contains enough column basis. From KSIR Algorithm, we can see the efficiency of KSIR in extracting the *e.d.r.* directions. The computational cost is one time of $O(\tilde{n}^3)$ for $\Sigma_{\tilde{K}}^{-1}$ and one time of $O(J^3)$ for the eigenvalue problem incurred, where $\tilde{n}$ ($\ll n$) is at most in the hundreds and J is at most in the tens in our later experiments.

4 Numerical Experiments

In this section, we will design numerical experiments to evaluate the information content about y contained in the *e.d.r.* subspace extracted by the proposed approximate KSIR algorithm. We focus on three kinds of applications of KSIR, namely data visualization, classification and regression. All experimental examples have two main steps: extracting the *e.d.r.* subspace and running a linear learning algorithm such as FDA, SVM, or regularized least squares (RLS) SVR on the *e.d.r.* variates. We evaluate the effectiveness of KSIR on five binary classification data sets, eight multi-class data sets and six regression data sets and compare the results with the conventional nonlinear SVM and SVR using the benchmark algorithm LIBSVM. All the experimental data sets are described in Tables 2 and 3. The R.S. columns record the proportion of random subset for reduced kernel approximation used in our experiments. The `banana` and `splice` data sets are from (Mika, Rätsch, Weston, & Schölkpf and K.-R. Müller, 1999). The `tree` data set is taken from Image Processing Lab, University of Texas at Arlington[1]. The `adult` and the `web` data sets are both compiled by Platt (Platt, 1999). The `medline`[2] is a text classification data set. One important characteristic of text classification data is the large number of variable dimensionality ($p \gg n$). Unlike the classical SIR, which works on $p \times p$ between-slice and total covariance matrices, our KSIR algorithm can easily handle this kind of data sets with $p \gg n$ by employing linear kernel and works on the $n \times n$ linear kernel data matrix without getting into the difficulty of large covariance matrices. We defer this special handling to Section 4.2. All the other data sets can be

[1] http://www-ee.uta.edu/EEweb/IP/training_data_files.htm

[2] The `medline` data set is available at http://www.cc.gatech.edu/∼hpark/data.html

Table 2: Description of classification data sets used in our experiments.

Data set	Classes	Training Size	Testing Size	Attributes	R.S. (%)
banana	2	400	4900	2	10%
tree	2	700	11692	18	10%
splice	2	1000	2175	60	10%
adult	2	32561	16281	123	1%
web	2	49749	14951	300	1%
Iris	3	150	-	4	10%
wine	3	178	-	13	10%
vehicle	4	846	-	18	20%
segment	7	2310	-	19	10%
dna	3	2000	1186	180	10%
satimage	6	4435	2000	36	20%
pendigits	10	7494	3498	16	4%
medline	5	1250	1250	22095	100%

obtained from UCI Repository of machine learning data archive (Asuncion & Newman, 2007) and UCI Statlog collection. The nonlinear kernel we used in all experiments is the radial basis function (Gaussian kernel). All our codes are written in Matlab (MATLAB, 1992) and are available at `http://www.stat.sinica.edu.tw/syhuang/`.

4.1 Data visualization

SIR and KSIR aim to extract a low dimensional *e.d.r.* subspace that contains the information about output variable y as much as possible. The former looks for such a subspace in the pattern Euclidean space, while the latter in the feature RKHS. Moreover, both SIR and KSIR algorithms rank the importance of *e.d.r.* directions by associated eigenvalues. Thus, we can use the first one or two directions to visualize the main data structure, which will be otherwise complex in high dimension. Here we show some data views in a 2-dimensional subspace obtained by PCA, SIR and KSIR, respectively. The

Table 3: Description of regression data sets used in our experiments.

Data set	Size	Attributes	R.S. (%)
`housing`	506	13	15%
`Comp_Activ_1000`	1000	21	5%
`Kin_fh_1000`	1000	32	5%
`Comp_Activ`	8129	21	5%
`Kin_fh`	8129	32	5%
`Friedman`	40768	10	1%

first example is on `pendigits` data. Training data are used to extract the *e.d.r.* directions, and only 14 test points from each category are used to plot the low-dimensional views to avoid excessive ink. The results are shown in Figure 1. Figures 1(a) and 1(b) are 2D views via PCA and SIR. There are some obvious overlaps among these ten classes. Figure 1(c) is the 2D view along KSIR directions and Figure 1(d) zooms in a small crowded region for a better view. We can easily see that KSIR variates provide a much better discriminant power.

The next two examples are on regression data sets. The first one is the "`peaks`" function provided by Matlab. It creates a synthetic regression surface (without additive noise) using 2-dimensional inputs. The corresponding regression surface plot with 961 points is in Figure 2(a). In our study, we split this data set into 80% and 20% subsets for training and testing, respectively. Plots of 2D views by PCA, SIR and KSIR are in Figures 2(b)-(d). As the `peaks` data have only 2-dimensional inputs, there is not much sense to talk about dimension reduction for 2D inputs. The purpose of this data example is to show the relevant linear structure in feature subspace by KSIR as depicted in Figure 2(d). We can clearly see that the KSIR processing turns the nonlinear structure in pattern space into a prominent linear structure in kernel feature space. It provides an empirical justification to combine the KSIR with linear learning algorithms for other tasks on the *e.d.r.* feature subspace.

Another regression example is the Friedman's example (5). There are ten explana-

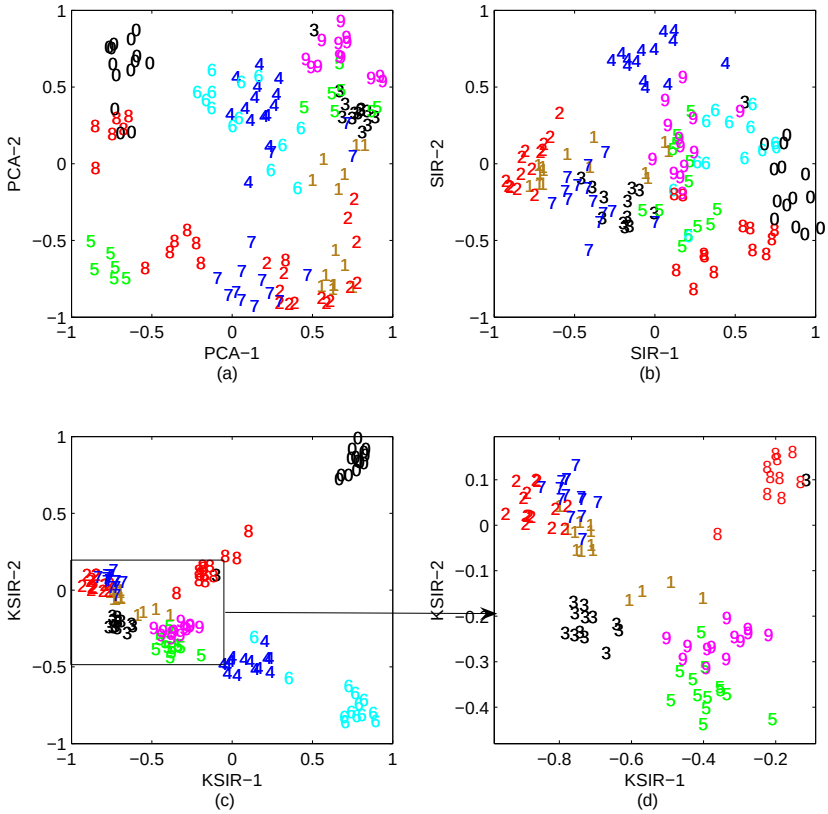

Figure 1: 2D views of `pendigits` data by PCA, SIR and KSIR.

tory variables, but only five of them are effective and the rest are nuisance. There are linear component, $10x_4 + 5x_5$, as well as nonlinear component, $\sin(\pi x_1 x_2)$, and a hidden *e.d.r.* direction x_3 to SIR in this example. We split the data into 99% and 1% subsets for training and testing, respectively, for data visualization purpose. The leading ten eigenvalues by SIR and KSIR are respectively

$$\text{SIR} : 0.7213, 0.0067, 0.0020, 0.0013, 0.0011, 0.0007, 0.0004, 0.0004, 0.0003, 0.0001;$$

$$\text{KSIR} : 0.9417, 0.6639, 0.2010, 0.0435, 0.0214, 0.0120, 0.0111, 0.0105, 0.0097, 0.0092.$$

The low-dimensional data views by PCA, SIR and KSIR are shown in Figure 3. Obviously, none of the PCA directions capture a good effective subspace for the response

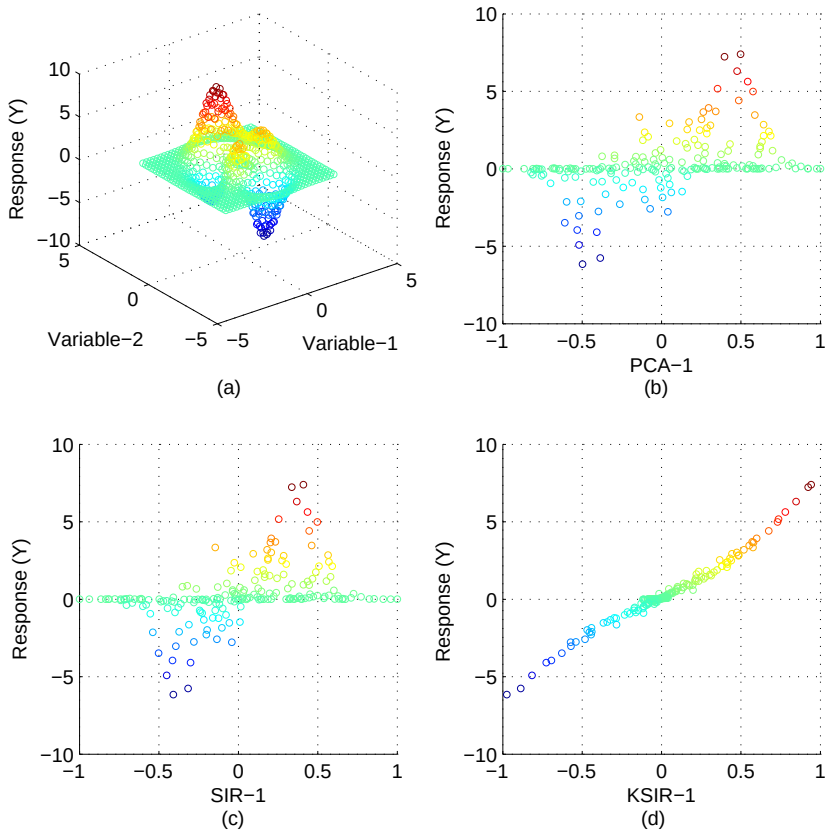

Figure 2: 2D views of response vs. the 1st variate by PCA, SIR and KSIR with peaks data.

surface. The first SIR direction does reflect a good description for the response, and the first KSIR direction is even better and it carries the best information content for the response among the three methods. Figure 3(c) shows a clearly good linearity of the first KSIR variate to the response. In summary, the effect of KSIR is not mere dimension reduction, it also maps the data to low-dimensional nonlinear features via kernel transform so that the response can be well approximated by a linear form in terms of these extracted features.

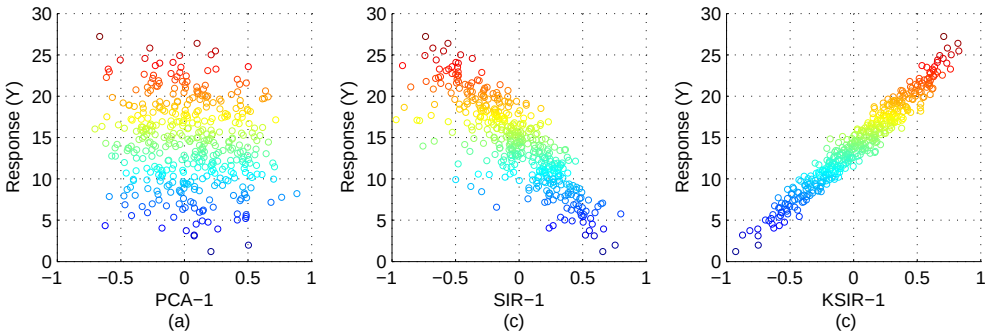

Figure 3: 2D views of Friedman data by PCA, SIR and KSIR.

4.2 KSIR dimension reduction for classification

The dimension reduction provided by KSIR can be used as a data preprocess for later task such as classification or regression. In the process of applying SIR or KSIR, the role of each slice represents the clustering structure of the data. If we know the information about clustering structure in advance, it helps us to make slices when applying SIR or KSIR. In classification, the clustering structure has been defined through their class labels, and slices are made accordingly. We then estimate the central *e.d.r.* subspace and map the data onto this subspace for discriminant purpose. In a J-class problem, we slice the data sets into J slices according to the class labels. Thus, there are at most $J-1$ many independent *e.d.r.* directions, since the rank of $\Sigma_{E(A|Y_J)}$ or $\Sigma_{E(K|Y_J)}$ is at most $J-1$. After extracting the *e.d.r.* subspace, discriminant analysis becomes much computationally easier in this very low-dimensional subspace. Since we have turned the nonlinear structure in the pattern space into an *approximately linear* structure in the feature space via kernel transformation, direct application of linear learning algorithms on KSIR variates is often sufficient. In our classification experiments, we particularly pick the Fisher linear discriminant analysis (FDA) and the linear smooth support vector machine (SSVM) (Lee & Mangasarian, 2001) as our baseline learning algorithms. One property of SSVM is that it is solved in the primal space and its computational complexity depends on the number of input attributes (here the number of KSIR variates). Smaller number of columns implies less computational load. Note that as data are

projected along the KSIR directions, discriminant analysis therein is computationally light. The FDA and SSVM acting on top of KSIR variates are numerically compared with the standard nonlinear SVM benchmark algorithm LIBSVM (Chang & Lin., 2001). One-versus-one scheme is used for SSVM and LIBSVM for multi-class problems.

For binary classification, the data sets put on experiments are already divided into training set and testing set in advance. We use the training set to build the model, including extracting the *e.d.r.* subspace and training for the final model in this *e.d.r.* subspace, then evaluate the resulting model using the testing set. As there is some stochastic variation due to reduced set selection, the procedure is repeated 10 times. The upper panel of Table 4 lists the average error rates for these binary classifications. Reduced kernel approximation by random subset with proportion 1% or 10% is used. For binary classification, KSIR extracts a one-dimensional feature component for discriminant analysis. FDA in this one-dimension subspace is a simple division of the line through the midpoint of two class centroids. The SSVM division of the line is a bit more complex than cutting through the midpoint. It is still a maximum margin criterion, but along a line instead of in a high-dimensional feature space. Results are compared with nonlinear LIBSVM. Although FDA and SSVM are acting on top of a one-dimension KSIR variate and reduced kernel approximation has been applied, the results in the KSIR columns are comparable to those in the LIBSVM column. It means that KSIR actually finds an effective projection direction in the kernel feature space and the reduced kernel approximation works well, too. For multi-class problems, the first four data sets are not pre-divided into training and testing sets. For these data sets, we use ten-fold cross validation and report their average error rates over 10 replicate runs of ten-fold CV. Results are listed in the middle panel of Table 4. For the rest multi-class data sets, there are separate testing sets. The results of ten replicate runs are reported in the lower panel of Table 4. Note that for the `medline` data set, the variable dimensionality is 22095. For this example we look for linear *e.d.r.* directions in the original pattern space instead of kernel *e.d.r.* directions. As the dimensionality is so high that a direct application

of the classical SIR, which calls for covariance matrices of size 22095, is not possible. Our KSIR algorithm can overcome this problem by using the linear kernel, $K = AA'$ (even the reduced linear kernel, $K = A\tilde{A}'$). In linear kernel setting, the computation cost is at most $O(n^3)$, where n is the sample size. This is the special handling for data sets with $n \ll p$, which is commonly seen in text mining and gene expression data sets. From Table 4, we see that a simple linear SSVM algorithm acting on a few leading KSIR variates can perform as good as nonlinear LIBSVM. The performance of FDA on KSIR variates is a little worse than SSVM on KSIR variates and nonlinear LIBSVM, but the difference is small.

Table 4: The average error rate for FDA and linear SSMV on KSIR variates compared with nonlinear LIBSVM on classification data sets.

Data set	KSIR+FDA	KSIR+SSVM	LIBSVM
banana	0.1170	0.1214	0.1228
tree	0.1234	0.1179	0.1283
splice	0.1292	0.1200	0.1012
adult	0.1671	0.1488	0.1491
web	0.0169	0.0149	0.0090
Iris	0.0213	0.0227	0.0380
wine	0.0131	0.0094	0.0181
vehicle	0.1468	0.1483	0.1429
segment	0.0309	0.0288	0.0283
dna	0.0659	0.0453	0.0460
satimage	0.0914	0.0904	0.0872
pendigits	0.0224	0.0188	0.0177
medline	0.1208	0.1136	0.1106

Another issue is the computing time comparison. We record all the computing time CPU seconds in Table 6. All the experiments are executed in the same environment. The equipment of the computer is CPU P4 3.0GHz, Memory 1.0G and operating system XP. It can be seen that KSIR+FDA and KSIR+SSVM are often faster than LIBSVM

in training time especially for large data sets and multi-class problems. For multi-class problems, we used "one-versus-one" scheme to decompose the problem into a series of binary classification subproblems and combine them by a majority vote. The KSIR+FDA and KSIR+SSVM only have to run the KSIR algorithm once, which consumes the major part of the computing time in one complete run of discriminant analysis. Once we have the *e.d.r* variates with dimensionality $J-1$, a series of binary SSVMs or one FDA machine in this $(J-1)$-dimensional subspace is computationally light to carry out. In comparing the hybrid of KSIR with linear SSVM versus the nonlinear LIBSVM, both have to build a series of C_2^J many binary classifiers. The former needs a one-time-only KSIR process in a reduced feature subspace of dimensionality $\tilde{n}$ and then builds a series of binary classifiers in a $(J-1)$-dimensional KSIR extracted subspace, while the latter builds such a series of binary classifiers in a higher dimensional feature space of dimensionality about the size $2n/J$. We only report the training time comparison with LIBSVM. The testing time for linear learning algorithms (FDA and SSVM here, and RLS-SVR in next subsection) on KSIR test variates are prominently and uniformly faster than LIBSVM in regression and in classification, and thus we omit the report of testing time comparison. The efficient testing time is due to the fact that KSIR-based learning algorithms are acting on very low-dimensional KSIR variates, while LIBSVM is acting in the high dimensional feature space and its speed depends on the number of support vectors, which is much larger than the *e.d.r.* dimensionality.

4.3 KSIR dimension reduction for regression

Different from classification problems, KSIR for regression can be more complicated than classification due to the lack of intuitive slices. In regression, we need to consider more factors, like the number of slices, their positioning and the dimensionality of the final *e.d.r.* subspace. For positioning of slices, we adopt a simple equal frequency strategy for it. For the number of slices, we fix at 30 slices in all our regression examples, because it performs reasonably well among a few quick trials of various numbers of

Table 5: The training time (seconds) of FDA and linear SSVM on KSIR variates compared with nonlinear LIBSVM on classification data sets.

Data set	KSIR+FDA	KSIR+SSVM	LIBSVM
banana	0.063	0.078	0.016
tree	0.141	0.078	0.078
splice	0.109	0.109	0.422
adult	6.032	6.110	255.631
web	37.374	37.406	174.190
dna	0.329	0.344	2.900
satimage	3.828	3.953	4.593
pendigits	1.390	2.058	2.953
medline	1.993	2.016	3.033

slices. It is also known that the results are quite robust to the slice number (Li, 1991). In determining the number of extracted components for final data analysis, we take two different numbers of dimensionality, 3 and 29, for simplicity. There are more rigorous statistical procedures based on output eigenvalues to determine the dimensionality of *e.d.r.* subspace (Cook, 1998; Ferré, 1998; Li, 1991). However, we will not pursue this issue here. The dimensionality 29 is the largest that we can go for 30 slices. In our empirical experience, the last few eigenvalues are relatively much smaller than the leading ones. Often a few leading directions, say 3, can be well enough for describing the response without losing much information.

After extracting the KSIR directions, we apply the linear regularized least square (RLS) fit of the responses on KSIR variates. The testing results of KSIR+RLS are compared with nonlinear SVR provided in LIBSVM. Note that a reduced kernel approximation has been used in KSIR for all our regression examples. The R^2 values based on ten-fold cross validation are shown in Table 6. R^2 is a commonly used criterion for

Table 6: R^2 of RLS on 3 and 29 KSIR variates compared with R^2 of nonlinear LIBSVM on regression data sets.

Data set	KSIR(3)+RLS	KSIR(29)+RLS	LIBSVM
`housing`	0.8543	0.8462	0.8687
`Comp_Activ_1000`	0.9685	0.9732	0.9776
`Kin_fh_1000`	0.6452	0.6482	0.6491
`Comp_Activ`	0.9760	0.9789	0.9820
`Kin_fh`	0.6964	0.6975	0.7014
`Friedman`	0.9556	0.9556	0.9559

evaluation of regression goodness of fit. Its definition is given below:

$$R^2 = 1 - \frac{\|y - \hat{y}\|^2}{\|y - \bar{y}\|^2},$$

where $\hat{y}$ is the fitted response and $\bar{y}$ is the grand mean. We can see that the difference in R^2 between KSIR(29)+RLS and nonlinear LIBSVM is not significant. Note that nonlinear LIBSVM is only a little better than KSIR(3)+RLS. In other words, KSIR+RLS is competent for regression even with only 3 components. Computing time comparison is reported in Table 7. KSIR+RLS is much faster than nonlinear LIBSVM, especially for large data problems. All these results reflect the same phenomenon found in classification problems, that KSIR can find effective *e.d.r.* directions to speed up the computation for regression fit with satisfactory results.

4.4 Parameter tuning

Throughout our experimental study, the Gaussian kernel $K(x, u) = \exp(-\gamma\|x - u\|^2)$ is used except for the `medline` data set. For either KSIR-based methods or the conventional SVMs, such as LIBSVM, there are two parameters involved, namely, the weight parameter C and the Gaussian kernel width parameter γ. The naive tuning procedure, a two-dimensional grid search, in conventional SVMs is time consuming. A remedy for it is

Table 7: The training time (seconds) of RLS on 3 and 29 KSIR variates compared with the training time of nonlinear LIBSVM on regression data sets.

Data set	KSIR(3)+RLS	KSIR(29)+RLS	LIBSVM
`housing`	0.022	0.040	0.211
`Comp_Activ_1000`	0.023	0.056	0.505
`Kin_fh_1000`	0.022	0.041	0.395
`Comp_Activ`	1.423	1.508	27.606
`Kin_fh`	1.416	1.502	20.950
`Friedman`	8.452	8.745	2400.1

to replace the bulldozer grid-search by a well-designed search scheme to reduce computing load. For instance, the nested uniform design (UD) model selection method (Huang, Lee, Lin, & Huang, 2007) provides an economic alternative for parameter tuning. It has been numerically shown that there is no significant difference in testing accuracy by using a nested-UD search to replace the grid search. Thus, in our experimental study, the nest-UD is adopted for parameter tuning in LIBSVM. As for KSIR-based methods, computing time for parameter tuning is a lot lighter than that for conventional SVMs, even if the latter is equipped with a UD-based search. Tuning procedure for KSIR-based methods is *nearly* a one-dimensional search for γ, as the search for C is computationally light and negligible. KSIR-based methods are carried out in two stages. At the first stage, a parameter value for γ is needed for training KSIR *e.d.r.* subspace. At the second, a parameter value for C is needed for linear SSVM or RLS on KSIR variates. For each γ and its resulting *e.d.r.* subspace, an optimal C is determined over a range of grid points. This tuning procedure at the second stage for C is computationally light, as it is carried out in a very low-dimensional *e.d.r.* subspace and the time complexities of linear SSVM and RLS depend on the dimensionality of the *e.d.r.* subspace. For each fixed γ we can try a few C values to pair with this γ without much computing cost. This is another computing merit of KSIR approach in practical usage.

5 Conclusion

We have introduced an effective nonlinear dimension reduction technique, KSIR, which kernelizes the classical SIR algorithm using the same notion of spectrum decomposition in a feature RKHS. The KSIR algorithm first maps the pattern data to an appropriate RKHS, and next extracts the main linear features in this embedded feature space. It takes class labels or regression response information into account and is a supervised dimension reduction method. After the extraction of the *e.d.r.* subspace, many supervised linear learning algorithms, such as FDA, SVM, SVR and possible others, can be applied to the images of input data in this *e.d.r.* feature subspace. This will generate a nonlinear learning model in the original input space and achieve a very good performance for complex data analysis. We have also incorporated reduced kernel approximation to cut down the computational load and to resolve the numerical instability due to singularity in between-slice covariance matrix. The singularity problem not only causes numerical instability but also leads to inferior *e.d.r.* directions estimation.

A few leading components extracted by KSIR can carry most of the relevant information about y in regression and in classification. It allows us to run linear learning algorithms in a very low dimensional *e.d.r.* subspace and to gain computational advantages without sacrificing the performance of learning algorithms. For example in solving nonlinear SVM multi-class problem, one has to solve C_2^J nonlinear binary SVMs under the "one-versus-one" scheme. In KSIR-based approach, it only involves solving the KSIR problem once, and the remaining task is solved by a series of C_2^J many linear binary SVMs in a $(J-1)$-dimensional space. Moreover, KSIR approach also has an advantage in tuning procedure. We have demonstrated these nice merits in our numerical experiments. Finally, using the first one or two components will help scientists or data analysts to gain a direct insight of data patterns, which will be otherwise complex in high dimension.

Appendix: KSIR in an RKHS Framework

The classical SIR looks for *e.d.r.* directions in the pattern Euclidean space for maximum between-slice dispersion with respect to overall dispersion. Based on the same idea of maximum between-slice dispersion, the kernel transform embeds the Euclidean pattern space $\mathcal{X}$ into an appropriate $\mathcal{H}_K$ by $\Gamma : \mathbf{x} \mapsto K(\mathbf{x}, \cdot)$. Next KSIR looks for *e.d.r.* directions in $\mathcal{H}_K$ that maximizes the between-slice dispersion with respect to the total dispersion. For technical details, we need to introduce a few more notations. Define below the grand mean function $m(\cdot)$, conditional mean function $m_{\mathbf{y}}(\cdot)$, covariance kernel Σ and between-slice covariance kernel Σ_B:

$$
\begin{aligned}
m(s) &= E\{K(\mathbf{x}, s)\}, && (17)\\
m_{\mathbf{y}}(s) &= E\left\{K(\mathbf{x}, s)|\mathbf{y}\right\}, && (18)\\
\Sigma(s,t) &= E\left\{(K(\mathbf{x}, s) - m(s))(K(\mathbf{x}, t) - m(t))\right\}, && (19)\\
\Sigma_B(s,t) &= E\left\{(m_{\mathbf{y}}(s) - m(s))(m_{\mathbf{y}}(t) - m(t))\right\}. && (20)
\end{aligned}
$$

Σ and Σ_B are also called covariance operators, as they induce linear operators on $\mathcal{H}_K$ given by

$$
(\Sigma f)(\cdot) = \int K(x, \cdot) f(x) dP_{\mathbf{x}}(x)
$$

and

$$
(\Sigma_B f)(\cdot) = \int E\left(K(\mathbf{x}, \cdot)|\mathbf{y}\right) E\left(f(\mathbf{x})|\mathbf{y}\right) dP_{\mathbf{y}}(y).
$$

KSIR solves the spectral decomposition of the between-slice covariance operator Σ_B with respect to the overall covariance operator Σ. That is, it aims to find leading directions $h_k \in \mathcal{H}_K$ to

$$
\text{maximize } \Sigma_B h_k = \lambda_k \Sigma h_k \text{ subject to } \langle h_k, \Sigma h_k \rangle_{\mathcal{H}_K} = \delta_{kj}, \qquad (21)
$$

where $\delta_{kk} = 1$ and $\delta_{kj} = 0$ for $k \neq j$. Below is a functional version of Theorem 1.

Theorem 1* *Assume the existence of an e.d.r. subspace* $\mathcal{H} = \text{span}\{h_1, \ldots, h_d\}$ *in* $\mathcal{H}_K$ *and the validity of the LDC (10). Further assume that the covariance operator* Σ *is*

compact and non-singular and that $m_{\mathbf{y}} - m$ *is in the range of* $\Sigma^{1/2}$. [3] *Then the central inverse regression curve* $m_{\mathbf{y}} - m$ *falls into the subspace* $\Sigma(\mathcal{H})$.

Proof of Theorem 1:* Note that for any $f, g \in \mathcal{H}_K$ we have (Janson, 1997)

$$\langle m, f\rangle_{\mathcal{H}_K} = E\langle K(\mathbf{x}, \cdot), f(\cdot)\rangle_{\mathcal{H}_K} = Ef(\mathbf{x}) \text{ and } \langle \Sigma f, g\rangle_{\mathcal{H}_K} = Cov\{f(\mathbf{x}), g(\mathbf{x})\}.$$

In particular, take $g = K(u, \cdot)$ and $f = h_k$, we have

$$\langle \Sigma h_k, g\rangle_{\mathcal{H}_K} = Cov\{h_k(\mathbf{x}), K(\mathbf{x}, u)\}. \tag{22}$$

Also the covariance of $h_i(\mathbf{x})$ and $h_j(\mathbf{x})$ is given by

$$Cov\{h_i(\mathbf{x}), h_j(\mathbf{x})\} = \langle \Sigma h_i, h_j\rangle_{\mathcal{H}_K}. \tag{23}$$

Let Σ_H denote the matrix with the (i, j)th entry given by (23) and let $H(\mathbf{x})$ be the random vector (column vector) given by $H(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_d(\mathbf{x}))'$. Note that

$$m_{\mathbf{y}}(\cdot) - m(\cdot) = E\left\{ E(K(\mathbf{x}, \cdot)|H(\mathbf{x})) - m(\cdot) \middle| \mathbf{y} \right\}.$$

By the LDC we have $E(K(\mathbf{x}, \cdot)|H(\mathbf{x})) - m(\cdot)$ is linear in $H(\mathbf{x})$, which leads to

$$E(K(\mathbf{x}, u)|H(\mathbf{x})) - m(u) = c(u)H(\mathbf{x}),$$

where $c(u) = Cov\{K(\mathbf{x}, u), H(\mathbf{x})'\}\Sigma_H^{-1}$ (a d-row vector) by the application of least squares linear regression. From (22) we have

$$c(u) = \langle \Sigma H(\cdot)', K(\cdot, u)\rangle_{\mathcal{H}_K}\Sigma_H^{-1} = (\Sigma H(u))'\Sigma_H^{-1}.$$

$\Sigma H(u)$ is independent of $\mathbf{y}$ and is in $\text{span}\{\Sigma h_1(u), \ldots, \Sigma_K h_d(u)\}$. □

Theorem 1* says that $(m_{\mathbf{y}} - m)$ falls into the subspace $\Sigma(\mathcal{H})$. In other words, $\Sigma^{-1}(m_{\mathbf{y}} - m)$ is in the *e.d.r.* subspace. Thus, orthonormalized $\Sigma^{-1}(m_{\mathbf{y}} - m)$ can be used for the estimation of *e.d.r.* directions. The idea is first to slice the y-variable into

[3] The compactness condition is to ensure that $\langle f, \Sigma g\rangle_{\mathcal{H}_K}$ is well-defined and the last condition on $m_{\mathbf{y}} - m$ is to ensure that $\langle m_{\mathbf{y}} - m, \Sigma^{-1}(m_{\mathbf{y}} - m)\rangle_{\mathcal{H}_K}$ exists and won't go unbounded.

J slices and denote the slice means by m_j, i.e., $m_j(\cdot) = E\{K(\mathbf{x}, \cdot)|\mathbf{y} \in j\text{th slice}\}$. Next is to orthonormalize $\Sigma^{-1}(m_j - m)$, $j = 1, \ldots, J$, and use them for *e.d.r.* directions. The idea can be formalized in the following proposition. Let M be a $J \times J$ matrix with (j, j')th entry given by

$$\underline{M} := \left[\sqrt{\pi_j \pi_{j'}}\, \langle m_j - m,\, \Sigma^{-1}(m_{j'} - m)\rangle_{\mathcal{H}_K}\right]_{jj'},$$

where π_j is the prior probability for the jth slice. Denote its decomposition by $M = UDU'$, where U consists of eigenvectors with non-zero eigenvalues and D is a diagonal matrix of non-zero eigenvalues. Let $W = (w_1, \ldots, w_J)$, J many functions arranged in row, where $w_j(u) = \sqrt{\pi_j}\,(m_j(u) - m(u))$ is the jth centered weighted slice mean.

Proposition 2* $\Sigma^{-1}WUD^{-1/2}$ *are e.d.r. directions.*

Proof of Proposition 2:* To show this proposition, we have to check that there exists a certain positive definite diagonal matrix Λ such that

$$\Sigma_B(\Sigma^{-1}WUD^{-1/2}) = \Sigma(\Sigma^{-1}WUD^{-1/2})\Lambda \quad \text{and}$$

$$(\Sigma^{-1}WUD^{-1/2})'\ \Sigma\ (\Sigma^{-1}WUD^{-1/2}) = I$$

hold. Note that Σ_B can be written as $\Sigma_B = WW'$. Then,

$$\Sigma_B(\Sigma^{-1}WUD^{-1/2}) = W\underline{W}'\underline{\Sigma}^{-1}\underline{W}UD^{-1/2} = WUD^{1/2}.$$

That is, take $\Lambda = D$. The derivation of the second assertion straightforwardly goes as follows:

$$D^{-1/2}U'W'\Sigma^{-1}WUD^{-1/2} = D^{-1/2}U'MUD^{-1/2} = I.$$

The proof is completed. □

In summary, KSIR is derived based on the same notion as SIR but in an infinite-dimensional RKHS. Its procedure is simply the SIR procedure on kernel transformed data.

Acknowledgment

The authors thank Dr. Wu, Han-Ming for helpful discussion.

References

Alpaydm, E. (2004). *Introduction to machine learning*. The MIT Press.

Asuncion, A., & Newman, D. J. (2007). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/mlrepository.html.

Chang, C. C., & Lin., C. J. (2001). LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Chen, C., & Li, K. C. (1998). Can SIR be as popular as multiple linear regression? *Statistica Sinica*, *8*, 289-316.

Cook, R. D. (1998). *Regression graphics: Ideas for studying regressions through graphics*. John Wiley and Sons.

Diaconis, P., & Freedman, D. (1984). Asymptotics of graphical projection pursuit. *Annals of Statistics*, *12*, 793-815.

Duan, N., & Li, K. C. (1991). Slicing regression: a link free regression method. *Annals of Statistics*, *19*, 505-530.

Ferré, L. (1998). Determining the dimension in sliced inverse regression and related methods. *Journal of American Statistical Association*, *93*, 132-140.

Friedman, J. H. (1991). Multivariate adaptative regression splines. *Annals of Statistics*, *19*, 1-67.

Hall, P., & Li, K. C. (1993). On almost linearity of low dimensional projection from high dimensional data. *Annals of Statistics*, *21*, 867-889.

Huang, C. M., Lee, Y. J., Lin, D. K. J., & Huang, S. Y. (2007). Model selection for support vector machines via uniform design. *A special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis*, *52*, 335-346.

Janson, S. (1997). *Gaussian hilbert spaces*. Cambridge: Cambridge University Press.

Lee, Y. J., & Huang, S. Y. (2007). Reduced support vector machines: a statistical theory. *IEEE Transactions on Neural Networks*, *18*, 1-13.

Lee, Y. J., & Mangasarian, O. L. (2001). SSVM: A smooth support vector machine. *Computational Optimization and Applications*, *20*, 5-22.

Li, K. C. (1991). Sliced inverse regression for dimension reduction (with discussion). *Journal of the American Statistical Association*, *86*, 316–342.

Li, K. C. (1997). Nonlinear confounding in high-dimensional regression. *Annals of Statistics*, *25*, 577-612.

MATLAB. (1992). *User's guide*. The MathWorks, Inc., Natick, MA 01760.

Mika, S., Rätsch, G., Weston, J., & Schölkpf and K.-R. Müller, B. (1999). Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX* (p. 41-48).

Platt, J. C. (1999). Sequential minimal optimization: a fast algorithm for training support vector machines. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning.* The MIT Press.

Saitoh, S. (1997). *Integral transforms, reproducing kernels and their applications*. Addison Wesley Longman.

Thompson, J. R., & Tapia, R. A. (1990). *Nonparametric function estimation, modeling, and simulation*. SIAM.

Wahba, G. (1999). Support vector machines, reproducing kernel Hillbert spaces, and randomized GACV. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning.* The MIT Press.

Wu, H. M. (n.d.). Kernel sliced inverse regression with applications on classification. *Journal of Computational and Graphical Statistics,* accepted, article available at http://idv.sinica.edu.tw/hmwu/Publications/index.htm.