Kernel SODA: A Feature Reduction Technique Using Kernel Based Analysis

Yinan Yu[a,b], Tomas McKelvey[a]
Signals and Systems
Chalmers University of Technology [a]
Gothenburg, Sweden
{*yinan; tomas.mckelvey*}*@chalmers.se*

S.Y. Kung[b]
Electrical Engineering
Princeton University [b]
Princeton, USA
{*yinany; kung*}*@princeton.edu*

***Abstract*—A feature extraction technique called Successively Orthogonal Discriminant Analysis (SODA) has been recently proposed to overcome the limitation of Linear Discriminant Analysis (LDA), whose objective is to find a projection vector such that the projected values of data from both classes have maximum class separability. However, in LDA, only one such vector can be found due to the rank deficiency for binary classification problems. On the other hand, as a feature extraction technique, the proposed algorithm SODA attempts to obtain a transformation matrix instead of a vector. In this paper, the kernel version of SODA is presented in both intrinsic space and empirical space. To obtain the solution without sacrificing numerical efficiency, we propose a relaxed formulation and data selection for large scale computations. Simulations are conducted on 5 data sets from UCI database to verify and evaluated the new approach.**

***Keywords*-Feature extraction, SODA, Kernel, Discriminant Analysis, big data**

I. Introduction

As a solution to the curse of dimensionality, feature reduction [1] for classification has been a popular topic for decades. There are many reasons why we should care about the dimensionality. 1) Overfitting is unevadable for high dimensional feature space, which might ruin the generalization ability of the classifier. 2) When the number of variables is too large, high storage capacity is required, and computational complexity is yet another issue. 3) In many cases, high dimensionality causes computational instability and singularity [2] 4) Class separability is very likely to be enhanced by eliminating redundant information.

There are two types feature reduction techniques: feature selection and feature extraction. Feature selection [3] is to select a subset of the variables with respect to some criteria. On the other hand, feature extraction attempts to find a function $f(x) : \mathbb{R}^m \rightarrow \mathbb{R}^k$, which transforms data from the original space $\mathbb{R}^m$ to a low dimensional feature space $\mathbb{R}^k$, where $m > k$. In this paper, we focus on the development of a feature extraction technique for classification. The proposed approach is the kernel extension of a previously presented technique called Successively Orthogonal Discriminant Analysis (SODA) [4]. The original technique is closely related to Linear Discriminant Analysis (LDA) [5] and Principal Component Analysis (PCA) [6].

The paper is organized as follows. Section II reviews the relation between LDA, PCA and SODA. Section III presents the formulations of Kernel SODA in both intrinsic space and empirical space. For simple numerical implementations, a relaxed KSODA algorithm is developed in Section IV. The experimental results are shown in the last section to verify the proposed technique.

II. Related Work

A. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) has a very long history. The underlying idea is based on Fisher criteria for maximizing the class separability.

Given class label $c \in \{+, -\}$ and training data $\mathcal{X}_c = \{\mathbf{x}_1^c, \cdots, \mathbf{x}_{N_c}^c\}$, the class separability is measured using the 'between-class scatter matrix' $\mathbf{S}_B$ and the 'within-class scatter matrix' $\mathbf{S}_W$, which are respectively defined as:

$$\begin{aligned}\mathbf{S}_W &= \frac{1}{N_+}\sum_{i=1}^{N_+}(\mathbf{x}_i^+ - \mu^+)(\mathbf{x}_i^+ - \mu^+)^T \\ &+ \frac{1}{N_-}\sum_{j=1}^{N_-}(\mathbf{x}_j^- - \mu^-)(\mathbf{x}_j^- - \mu^-)^T \\ \mathbf{S}_B &= (\mu^+ - \mu^-)(\mu^+ - \mu^-)^T \end{aligned} \quad (1)$$

where μ^+ and μ^- are the mean vector estimated from the corresponding class $+$ and $-$. In LDA, we would like to find a vector $\mathbf{w}$ that maximizes the following Fisher score:

$$J = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}, \quad (2)$$

and the solution vector is the generalized eigenvector corresponding to the largest generalized eigenvalue of the problem $\mathbf{S}_B\mathbf{w} = \mathbf{S}_W\mathbf{w}\lambda$. However, problems occur when the within matrix $\mathbf{S}_W$ is singular. There are many ways of tackling this problem [7], [8] and one way is to compute the eigenvector of the Fisher matrix $\mathbf{F} = \mathbf{S_W}^+\mathbf{S_B}$. We will adopt this method in this paper.

B. Principal Component Analysis (PCA)

One of the most famous dimensionality reduction techniques is the Principal Component Analysis (PCA), which finds the subspace with the largest variation. PCA can be formulated in an iterative fashion. Namely, we are trying to find a matrix $\mathbf{W} = \left[\mathbf{w}_1, \cdots, \mathbf{w}_k\right]$ whose columns satisfy:

$$\begin{aligned} \underset{\mathbf{w}_i}{\text{maximize}} \quad & \mathbf{w}_i^T \mathbf{S} \mathbf{w}_i \\ \text{subject to} \quad & \mathbf{w}_i \perp \mathbf{w}_{1,\ldots,i-1} \\ & \mathbf{w}_i^T \mathbf{w}_i = 1 \\ & \mathbf{w}_i \in \text{Span}(\mathbf{S}) \end{aligned} \tag{3}$$

where $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ is the covariance matrix of the training data and $\text{Span}(\mathbf{S})$ denotes the range space of matrix $\mathbf{S}$. When the dimensionality of the data vector is high, PCA can be implemented iteratively without computing the covariance matrix $\mathbf{S}$ [9]. The solution of PCA provides a transformation matrix that projects the data to a low dimensional subspace, where the data representation is enhanced. The data vector in the principal component space is thus represented as:

$$\mathbf{x} \to \mathbf{W}^T \mathbf{x}. \tag{4}$$

C. Successively Orthogonal Discriminant Analysis

Successively Orthogonal Discriminant Analysis (SODA) integrates the ideas of LDA and PCA. Defined as a successive approach, SODA adopts the Fisher discriminant score as its objective function for each iteration. The output of SODA is a transformation matrix that projects the data onto a new feature space with enhanced class separability. Similarities and differences of LDA, PCA and SODA can be found in Table I. The formulation of SODA is illustrated as follows.

SODA formulation. *The matrix* $\mathbf{W} = \left[\mathbf{w}_1 \cdots \mathbf{w}_k\right]$ *defines a map* $\mathbb{R}^m \to \mathbb{R}^k$*, whose columns satisfy:*

$$\begin{aligned} \underset{\mathbf{w}_i}{\textit{maximize}} \quad & \frac{\mathbf{w}_i^T \mathbf{S}_B \mathbf{w}_i}{\mathbf{w}_i^T \mathbf{S}_W \mathbf{w}_i} \\ \textit{subject to} \quad & \mathbf{w}_i \perp \mathbf{w}_{1,\ldots,i-1} \\ & \mathbf{w}_i^T \mathbf{w}_i = 1 \\ & \mathbf{w}_i \in \textit{Span}(\mathbf{S}_W) \end{aligned} \tag{5}$$

where $\textit{Span}(\mathbf{S}_W)$ *denotes the range space of matrix* $\mathbf{S}_W$*.* □

Similarly, the resulting data vector is mapped to a reduced feature space:

$$\mathbf{x} \to \mathbf{W}^T \mathbf{x}. \tag{6}$$

SODA and PCA share the same concept of optimal subspace projection. PCA's optimality enjoys the advantage of having a global criterion, however it does not take into account of teacher's information. On the other hand, SODA takes full account of teacher's information, but its optimality is formulated and executed step by step. Both PCA and LDA have their kernelized variance called KPCA and KDA. This motivates us to extend SODA to its kernel model.

The existing feature reduction techniques based on discriminant analysis [11] mostly depend on the number of classes C. For binary classification problem, i.e. $C = 2$, due to rank deficiency, a transformation matrix can not be found. On the other hand, SODA overcomes this limitation by modifying the searching space. More details can be found in later sections.

III. Theory of kernel SODA

In kernel based analysis, we define a function $\varphi(\mathbf{x}) : \mathbf{x} \to \varphi$, which maps the data vector from the original space to intrinsic space φ to obtain a better separation.

A. KSODA in Intrinsic Space

KSODA can be formulated in the intrinsic space with an explicit expression of $\varphi(\mathbf{x})$. The transformation matrix is denoted by $\mathbf{U} = \left[\mathbf{u}_1, \cdots, \mathbf{u}_k\right]$ in the intrinsic space.

Let $c \in \{+, -\}$ denote the class label and N_c the total training size of class c, we can define the between-class scatter matrix $\mathbf{S}_B^\varphi$ and within-class scatter matrix $\mathbf{S}_W^\varphi$ in the intrinsic space as follows:

$$\mathbf{S}_B^\varphi = (\mathbf{m}_+^\varphi - \mathbf{m}_-^\varphi)(\mathbf{m}_+^\varphi - \mathbf{m}_-^\varphi)^{\mathrm{T}} \tag{7}$$

$$\mathbf{S}_W^\varphi = \sum_{c \in \{+,-\}} \frac{1}{N_c} \sum_{j=1}^{N_c} (\varphi_j^c - \mathbf{m}_c^\varphi)(\varphi_j^c - \mathbf{m}_c^\varphi)^{\mathrm{T}}, \tag{8}$$

where

$$\mathbf{m}_c^\varphi = \frac{1}{N_c} \sum_{j=1}^{N_c} \varphi(\mathbf{x}_j^c). \tag{9}$$

KSODA (intrinsic space). *The matrix* $\mathbf{U} = \left[\mathbf{u}_1 \cdots \mathbf{u}_k\right]$ *defines a map* $\mathbb{R}^J \to \mathbb{R}^k$*, whose columns satisfy:*

$$\begin{aligned} \underset{\mathbf{u}_i}{\textit{maximize}} \quad & \frac{\mathbf{u}_i^T \mathbf{S}_B^\varphi \mathbf{u}_i}{\mathbf{u}_i^T \mathbf{S}_W^\varphi \mathbf{u}_i} \\ \textit{subject to} \quad & \mathbf{u}_i \perp \mathbf{u}_{1,\ldots,i-1} \\ & \mathbf{u}_i^T \mathbf{u}_i = 1 \\ & \mathbf{u}_i \in \textit{Span}(\mathbf{S}_W^\varphi) \end{aligned} \tag{10}$$

where $\textit{Span}(\mathbf{S}_W^\varphi)$ *denotes the range space of matrix* $\mathbf{S}_W^\varphi$*.* □

In this case, the data vector is transformed in the following way:

$$\mathbf{x} \to \varphi(\mathbf{x}) \to \mathbf{U}^T \varphi(\mathbf{x}). \tag{11}$$

Note that after the transformation $\mathbf{x} \to \varphi(\mathbf{x})$, the SODA algorithm can be applied directly.

	LDA	PCA	SODA
Type	Classifier	Feature Reduction	Feature Reduction
Output	$\mathbf{w}_{m\times 1}$	$\mathbf{W}_{m\times k}$	$\mathbf{U}_{m\times k}$
Purpose	Enhance Class Separability	Enhance Data Description	Enhance Class Separability
Objective Function	Direct	Direct	Successive
Implementation	Direct	Direct / Iterative	Iterative

Table I
THE RELATION BETWEEN LDA, PCA AND SODA.

B. KSODA in Empirical Space

When the Gaussian RBF kernel is adopted, the dimension of the intrinsic space is infinity and hence computations can not be carried out directly. In this case, it is necessary to resort to a kernel learning model proposed below.

From the theory of Reproducing Kernel Hilbert Space (RKHS) [12], [13], [14], each vector $\mathbf{u}_i \in \mathcal{H}$ can be written as a linear combination of all the training data $\varphi_1, \cdots, \varphi_N$ drawn from $\mathcal{H}$, i.e., we have

$$\mathbf{u}_i = \sum_{j}^{N} \varphi_j \mathbf{a}_{i,j} = \mathbf{\Phi}\mathbf{a}_i \tag{12}$$

where $\mathbf{\Phi} = \left[\varphi_1, \cdots, \varphi_N\right]$ is the training data matrix, the scalar $\mathbf{a}_{i,j}$ is the j^{th} element of vector $\mathbf{a}_i$. Similar to the SODA formulation, vector $\mathbf{u}_i$ is the i^{th} column of the transformation matrix $\mathbf{U}$ that can be written as

$$\mathbf{U} = \mathbf{\Phi}\mathbf{A} \tag{13}$$

where $\mathbf{A} = \left[\mathbf{a}_1, \cdots, \mathbf{a}_N\right]$

As a result of plugging Equation (12) into Equation (10), define matrices $\mathbf{M}$ and $\mathbf{N}$ [10]:

$$\mathbf{M} = (\mathbf{M}_+ - \mathbf{M}_-)(\mathbf{M}_+ - \mathbf{M}_-)^T. \tag{14}$$

and

$$\mathbf{N} = \sum_{c\in\{+,-\}} \mathbf{K}_c(\mathbf{I} - \frac{1}{N_c}\mathbf{E})\mathbf{K}_c^T, \tag{15}$$

where the row vectors of $\mathbf{M}_c$ are written as $(\mathbf{M}_c)_j = \frac{1}{N_c}\sum_{t=1}^{N_c} k(\mathbf{x}_j, \mathbf{x}_t^c)$. Matrix $\mathbf{K}_c$ denotes the kernel matrix $\mathbf{K}_c = \mathbf{\Phi}^T\mathbf{\Phi}_c$ and $\mathbf{E}$ is a matrix with all ones as its entries. This leads to an equivalent Kernel SODA learning model in the empirical space.

Denote $k(\mathbf{x}) = \mathbf{\Phi}^T\varphi(\mathbf{x})$, we formulate KSODA in the empirical space as:

KSODA (empirical space). *Find optimal vectors* $\mathbf{a}_1 \cdots \mathbf{a}_k$, *such that:*

$$\begin{aligned} \underset{\mathbf{a}_i}{\textit{maximize}} \quad & \frac{\mathbf{a}_i^T\mathbf{M}\mathbf{a}_i}{\mathbf{a_i}^T\mathbf{N}\mathbf{a}_i} \\ \textit{subject to} \quad & \mathbf{a}_i^T\mathbf{K}\mathbf{a}_j = 0, \quad \forall i \neq j \\ & \mathbf{a}_i^T\mathbf{K}\mathbf{a}_i = 1 \end{aligned} \tag{16}$$

The optimal transformation matrix $\mathbf{U}$ *can be written as:*

$$\mathbf{U} = \mathbf{\Phi}\left[\mathbf{a}_1 \cdots \mathbf{a}_k.\right]$$

□

Therefore, in the empirical space, the transformation follows:

$$\mathbf{x} \to k(\mathbf{x}) \to \mathbf{A}^T k(\mathbf{x}). \tag{17}$$

Equivalence: The empirical formulation is a direct result of kernelization of the intrinsic formulation [15], whose equivalence is straightforward i.e. $\mathbf{U}^T\varphi(\mathbf{x}) = \mathbf{A}^T k(\mathbf{x})$.

IV. IMPLEMENTATION AND APPROXIMATION

For user's choice, we describe two variance of kernelized SODA learning models.

A. KSODA implementation

There are two major steps involved in the algorithm:

- Initialization:

$$\mathbf{Q}^{(0)} = \mathbf{N}^+, \quad \mathbf{D}^{(0)} = \mathbf{K}^+, \quad \mathbf{\Delta} = \mathbf{M}_+ - \mathbf{M}_- \tag{18}$$

- Step 1: Computing $\mathbf{a}_i$ and normalization

$$\mathbf{a}_i = \frac{\mathbf{Q}^{(i)}\mathbf{\Delta}}{\sqrt{\mathbf{\Delta}^T\mathbf{Q}^{(i)T}\mathbf{K}\mathbf{Q}^{(i)}\mathbf{\Delta}}} \tag{19}$$

- Step 2: Let $\mathbf{D}^{(i+1)} = \mathbf{D}^{(i)} - \mathbf{a}_i\mathbf{a}_i^T$, update $\mathbf{Q}^{(i+1)}$ according to

$$\mathbf{Q}^{(i+1)} = \mathbf{D}^{(i+1)}\mathbf{N}^+\mathbf{D}^{(i+1)} \tag{20}$$

- Go to Step 1 until $i = k$.

B. Approximation

For numerical efficiency, we introduce two approximation strategies:

- Relaxation on orthogonality condition.
- Data selection for big data scenario and the purpose of numerical invertibility.

They are elaborated below.

1) Relaxation on orthogonality:: Due to the complexity of the weighted orthogonality constraint $\mathbf{A}^T\mathbf{K}\mathbf{A} = \mathbf{I}$, we define a relaxed formulation called KSODA (II).

KSODA (II). *Find optimal vectors* $\mathbf{a}_1 \cdots \mathbf{a}_k$*, such that:*

$$\begin{aligned} \underset{\mathbf{a}_i}{\textit{maximize}} \quad & \frac{\mathbf{a}_i^T\mathbf{M}\mathbf{a}_i}{\mathbf{a_i}^T\mathbf{N}\mathbf{a}_i} \\ \textit{subject to} \quad & \mathbf{a}_i^T\mathbf{a}_j = 0, \quad \forall i \neq j \\ & \mathbf{a}_i^T\mathbf{a}_i = 1 \\ & \mathbf{a}_i \in \textit{Span}(\mathbf{N}) \end{aligned} \tag{21}$$

where $\textit{Span}(\mathbf{N})$ *denotes the range space of matrix* $\mathbf{N}$ *and the transformed data vector is represented as:*

$$\mathbf{x}' = \left[\mathbf{a}_1 \cdots \mathbf{a}_k\right]^T k(\mathbf{x}). \tag{22}$$

□

The optimization problem stated in (21) can be solved by Algorithm KSODA(II). Similar proof can be found in [4].

Algorithm KSODA(II)

- Construct the matrix $\mathbf{X} = \left[\mathbf{x}_1, \cdots, \mathbf{x}_N\right]$. Define output dimension k.
- Compute $\mathbf{M}$ and $\mathbf{N}$ from Eq. (14) and (15) respectively,
- Let $\mathbf{N}^{(0)} = \mathbf{N}$

  $\mathbf{F}^{(1)} = (\mathbf{N}^{(0)})^+\mathbf{M}$
- For $i = 1 : k$
  - Solve for $\mathbf{F}^{(i)}\mathbf{a}_i = \lambda_i\mathbf{a}_i$

    where λ_i is the largest and only eigenvalue of $\mathbf{F}^{(i)}$.
  - Let $\mathbf{D}^{(i)} = \mathbf{I}_{m\times m} - \mathbf{a}_i\mathbf{a}_i^T$ be the deflation matrix

    $\mathbf{N}^{(i)} = \mathbf{D}^{(i)}\mathbf{N}^{(i-1)}\mathbf{D}^{(i)}$

    $\mathbf{F}^{(i+1)} = (\mathbf{N}^{(i)})^+\mathbf{M}$
- Form matrix: $\mathbf{A} = \left[\mathbf{a}_1 \cdots \mathbf{a}_k\right]$
- Transformation of the features: $\mathbf{x}' = \mathbf{A}^T\mathbf{K}(\mathbf{X}, \mathbf{x})$

C. Data selection and numerical invertibility

From the RKHS theories, the solution vectors $\mathbf{u}_i$ can be written as a linear combination of all the training data, namely $\mathbf{u}_i = \mathbf{\Phi}\mathbf{a}_i$. That means the size of the kernel matrix we have to compute is in the order of the training size N. Furthermore, the solution of Formulation KSODA is based on the pseudo-inverse of a $N \times N$ matrix $\mathbf{N}$, which results in a N^3 computational complexity.

To tackle this problem, we choose a subset $G \subset \Phi$ to approximate the span of the whole training space. We call such matrix $\mathbf{G}$ the basis matrix. Note that without ambiguity, we use capital letter for the set of some training patterns (e.g. $\Phi = \{\varphi_1, \cdots \varphi_N\}$) and boldface letter for the corresponding matrix (e.g. $\mathbf{\Phi} = \left[\varphi_1, \cdots, \varphi_N\right]$).

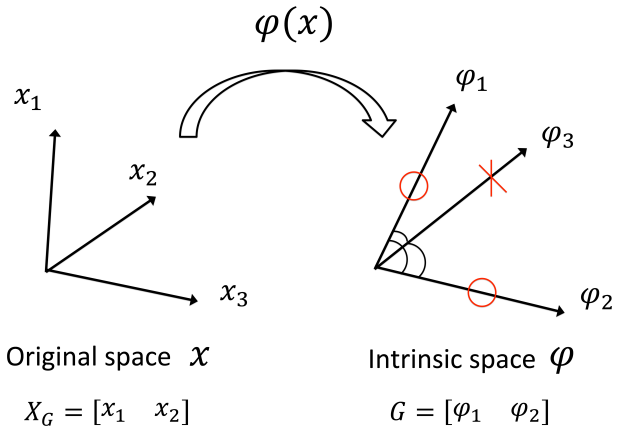

Figure 1. An intuitive illustration of the basis selection. If two vectors φ_i and φ_j are very 'similar' to each other, we assume that the spaces they span are collinear and there is no point to include both of them into the basis matrix $\mathbf{G}$. The similarity measure is naturally defined by normalizing the kernel function $\frac{k(\mathbf{x}_i,\mathbf{x}_j)}{\sqrt{k(\mathbf{x}_i,\mathbf{x}_i)}\sqrt{k(\mathbf{x}_j,\mathbf{x}_j)}}$. By excluding similar vectors, we could reduce the number of vectors in the basis matrix.

The idea is described as follows: *In the intrinsic space, given normalized vectors* φ_t*,* $t = 1, \cdots, N$*, we would like to find a basis matrix* $\mathbf{G} = \left[\varphi_1, \cdots \varphi_n\right]$*, such that for some small number* η*,* $\mathbf{G}^T\mathbf{G} = \mathbf{\Gamma}$*, where* $\mathbf{\Gamma}$ *is a full rank matrix and* $\frac{\Gamma_{i,j}}{\Gamma_{l,l}} < \eta, \forall l, i, j$ *with* $i \neq j$*.*

An illustrative example can be found in Figure IV-C. As we know that similarities between vectors always cause singularity and hence numerical instability. Therefore, by finding such basis matrix $\mathbf{G}$, the robust invertibility of the matrix $\mathbf{N}$ is enhanced.

Practically, the idea can be implemented as shown in Algorithm Basis $\mathbf{G}$.

V. Experimental Results

Parameter setting: In this section, we conduct simulations on 5 UCI [16] data sets to compare the classification results using original space, PCA, Kernel PCA, SODA, KSODA and KSODA (II). The classifier we used is Support Vector Machine (SVM) [17] with rbf kernel ($\sigma = 1$). We also compared the classification results with LDA and Kernel LDA. The parameter σ for the kernel based methods are set to be 0.5 for consistent and fair comparison. The reduced dimensionality for the feature reduction techniques under comparison is $k = 4$. This is chosen based on cross validation on the data set arcene and then applied to the rest of the data sets.

Data set		Dimension	Original	PCA	KPCA	LDA	KLDA	SODA	KSODA	KSODAII
arcene		$10^4(4) \times 200$	50%	21.67 %	18.75 %	37.48%	13.81%	31.99%	15.62 %	13.63 %
sonar		$60(4) \times 208$	27.52%	33.89 %	35.83 %	27.83%	19.88%	25.42%	20.38 %	17.43 %
wdbc		$30(4) \times 259$	9.17%	7.35 %	5.38 %	3.84%	7.18%	3.44%	2.44 %	2.36%
vehicle	van	$18(4) \times 199$	3.27%	12.23 %	17.86 %	2.01%	8.42%	2.39%	1.83 %	1.51%
	saab	$18(4) \times 219$	17.38%	26.20 %	28.19%	13.55%	18.19%	13.97%	12.16 %	10.55%
	bus	$18(4) \times 218$	2.23%	11.78 %	10.40%	3.21%	2.39%	3.10%	1.92 %	1.21%
	opel	$18(4) \times 212$	17.28%	27.53 %	27.48 %	13.04%	14.43%	12.99%	11.67 %	10.34%
segment	brickface	$19(4) \times 330$	0.87%	6.91%	5.96%	0.67%	1.05%	0.62%	0.59 %	0.53%
	sky	$19(4) \times 330$	0.25%	0.47%	0.26%	0.21%	1.78%	0%	0 %	0.01%
	foliage	$19(4) \times 330$	2.94%	7.69%	9.68%	4.03%	4.91%	3.28%	2.08 %	2.05 %
	cement	$19(4) \times 330$	1.74%	8.09%	7.46%	3.65%	4.89%	1.83%	1.45 %	1.45%
	window	$19(4) \times 330$	3.83%	10.20%	9.83%	4.34%	6.77%	3.66%	3.12 %	2.46%
	path	$19(4) \times 330$	0.42%	3.31%	5.73%	0.60%	1.01%	0.56%	0.33 %	0.32%
	grass	$19(4) \times 330$	0.58%	0.37%	0.42%	0.42%	0.21%	0.39%	0.21 %	0.19%

Table II
CLASSIFICATION ERROR COMPARISON BETWEEN DIFFERENT FEATURE REDUCTION TECHNIQUES. NOTE THAT THE KSODA IS ALREADY SUBSTANTIAL BETTER THAN SODA. HOWEVER, KSODA (II) OFFERS NOTICEABLE FURTHER IMPROVEMENT OVER KSODA.

There are another two parameters to choose for Kernel SODA algorithms, which are the size of the basis matrix $\mathbf{G}$ and the tolerant ratio η. These selections depend on the capacity of the computational device, the kernel parameters and the data properties. Here, we choose $N_G = 100$ and $\eta \in [0.5, 1)$ is selected for each data set by cross-validation.

Data description: The data sets we used are arcene, sonar, wdbc, vehicle and segment. The basic properties of the data sets are summarized in Table II.

Vehicle and segment have more than two classes. Since we focus on the study of binary classification in this paper, the results shown are based on the averaged error rate of one-versus-one scheme for all classes from the data sets.

Testing method: We divide each data set randomly into training set (80%) and testing set (20%). This procedure is repeated for 10 times. Furthermore, at the first time of the tests, 20% of the training set is left out for cross validation. The classification results in terms of error probabilities shown in Table II are based on the average error of the two classes:

$$P_{\text{err}} = \frac{1}{2} \sum_{c} \frac{\#\text{ of misclassified testing data in class } c}{\#\text{ of testing data in class } c \in \{+, -\}} \tag{25}$$

Testing results: On 13 out of 14 data sets, KSODA(II) has achieved the best classification results in terms of the averaged classification error defined in Equation (25). First, the original space of data set arcene has 10000 variables. It is obvious that it suffers from overfitting using SVM with rbf kernel, which results in a 50% error probability. In such scenarios, PCA/KPCA with extremely low dimensionality will do an even better job than the original space. However, when the number of original variables is reasonably small compared to the number of samples, PCA/KPCA do not achieve a better performance in general. LDA outperforms SVM on original space for roughly 50% cases. Since the parameter selection of kernel SVM is a key for high performance, LDA enjoys the advantage of simplicity.

On average, SODA/Kernel SODA algorithms give the

Algorithm Basis G

- Let $\mathbf{X} = \left[\mathbf{x}_1, \cdots, \mathbf{x}_N\right]$, construct an empty $m \times N_G$ matrix $\mathbf{X}_G$. Choose a small number η as the threshold. Choose a kernel function K and the size of the basis N_G.

  Set counter $k = 1$.
- for $i = 1 : N_G$

  Let $\mathbf{X}_G(:, i) := \mathbf{X}(:, k)$ $(\star)$
  - for $j = k + 1 : N$

    For given kernel function K, compute:

$$\mathbf{K} = K(\mathbf{X}_G, \mathbf{x}_k) \tag{23}$$

    Normalization:

$$\mathbf{K'}_{ij} = \frac{\mathbf{K}_{ij}}{\|\varphi(\mathbf{x}_i)\|_2 \|\varphi(\mathbf{x}_j)\|_2}, \quad \forall i, j \tag{24}$$

    where $\|\varphi(\mathbf{x}_t)\|_2 = \sqrt{K(\mathbf{x}_t, \mathbf{x}_t)}$ is the norm of vector φ_t $(t < i)$ on intrinsic space.
    - if: $\frac{\mathbf{K'}_{ij}}{\min(\mathbf{K}_{tt})} < \eta$, $\forall i \neq j, t < i$,
      let $k = j$ and return to $(\star)$.
    else: let $k = k + 1$.
  - end
- end
- Replace the matrix $\mathbf{X}$ in Algorithm KSODA(II) by $\mathbf{X}_G$.

best classification accuracy. The reason is that they do not only take into consideration of the class separability on the best one dimensional subspace, but also extend the development on k dimensions, which allow high flexibility that LDA does not provide. Kernel SODA, on the other hand, uses the kernel trick to further explore the intrinsic non-linear structure in the data and therefore results in an even lower classification error rate. Examples of visualization for these feature reduction techniques on data sets sonar, wdbc, vehicle and segment can be found in Figure 1 and 2.

VI. Acknowledgment

This material is based on research sponsored by DARPA under agreement number FA8750-12-2-0126. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

This work is also sponsored by the Swedish Research Council (VR) which is gratefully acknowledged.

References

[1] Fukumizu, K., Bach, F. R. and Jordan, M. I., *Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces.* The Journal of Machine Learning Research, Vol. 5, pp. 73-99, Jan. 2004.

[2] Hastie, T., Tibshirani, R., and Friedman J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Second Edition, Springer, Feb. 2009

[3] Guyon I. and Elisseeff A., *An introduction to variable and feature selection.* The Journal of Machine Learning Research, vol. 3, pp. 1157-1182, March 2003.

[4] Yu Y., Mckelvey T. and Kung S.Y., *A Classification Scheme for 'High-Dimensional-Small-Sample-Size' Data Using SODA and Ridge-SVM with Microwave Measurement Applications.* Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada, May 2013.

[5] Duda, R.O., Hart, P.E. and Stork, D.G., *Pattern Classification*, 2nd Edition, John Wiley & Sons, New York, 2011.

[6] Jolliffe I.T., *Principal Component Analysis.* Series: Springer Series in Statistics, 2nd ed., Springer, 2002.

[7] Hoerl A. E. and Kennard R. W. , *Ridge Regression: Biased Estimation for Nonorthogonal Problems.* Technometrics, vol. 12, No. 1, pp. 55-67 Feb., 1970.

[8] Yang J. and Yang J., *Why can LDA be performed in PCA transformed space?* Pattern Recognition, vol. 36, no. 2 , pp. 563-566, Feb. 2003.

[9] Diamantaras K. I., Kung S.Y., *Principal component neural networks: theory and applications.* John Wiley & Sons, 1996.

[10] Mika S., Ratsch G., Weston J., Scholkopf B., and Mullers K. R., *Fisher discriminant analysis with kernels.* Proceedings of the IEEE Signal Processing Society Workshop in Neural Networks for Signal Processing IX, pp. 41 - 48, Aug 1999

[11] Nie F., Xiang S., Liu Y., Hou C., and Zhang C., *Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction.* Pattern Recognition Letters, vol. 33, pp. 485491, 2012.

[12] Slavakis K., Theodoridis S., Yamada I., *Online classification using kernels and projection-based adaptive algorithms.* IEEE Transactions on Signal Processing, vol. 56(7), pp. 2781-2797, 2008.

[13] Bouboulis P. and Theodoridis S., *Extension of Wirtinger Calculus to Reproducing Kernel Hilbert Spaces and the complex kernel LMS.* IEEE Transactions on Signal Processing, vol. 53(3), pp. 964-978, 2011.

[14] Slavakis K., Bouboulis P., Theodoridis S., *Online Learning in Reproducing Kernel Spaces.* E-reference for Signal Processing, Elsevier, 2013.

[15] Kung S.Y., *Kernel Methods and Machine Learning.* Cambridge University Press, 2013.

[16] http://archive.ics.uci.edu/ml/.

[17] Mavroforakis M., Theodoridis S., *A Geometric Approach to Support Vector Machine (SVM) Classification.* IEEE Transaction on Neural Networks, vol. 17(3), pp.671-683, 2006.

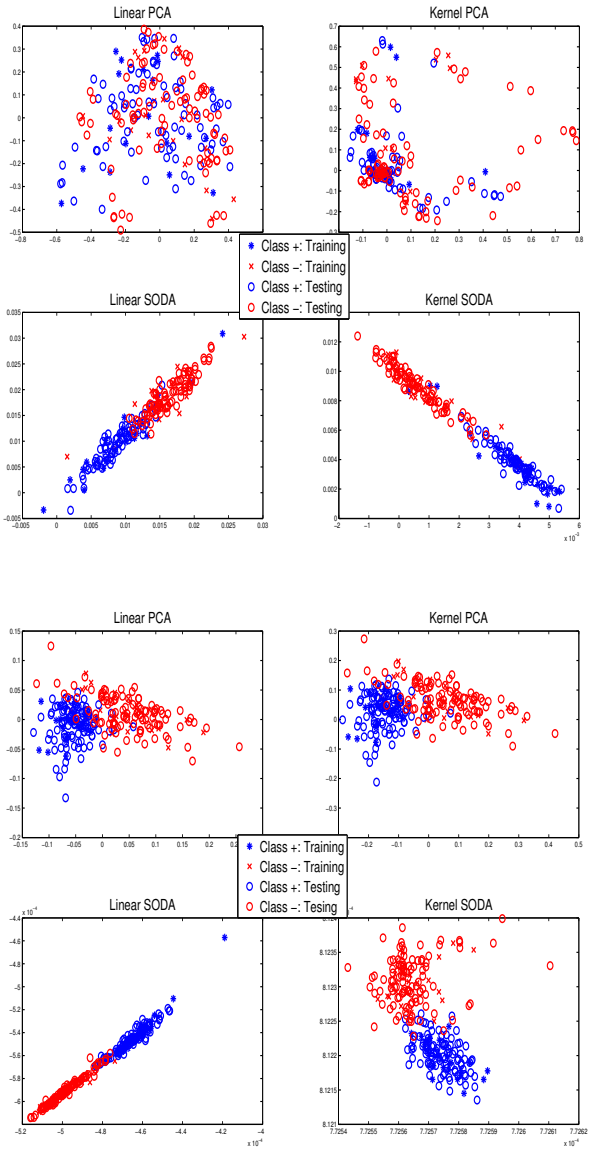

Figure 2. Visualization of the first two components of data set sonar and wdbc using different feature reduction techniques.

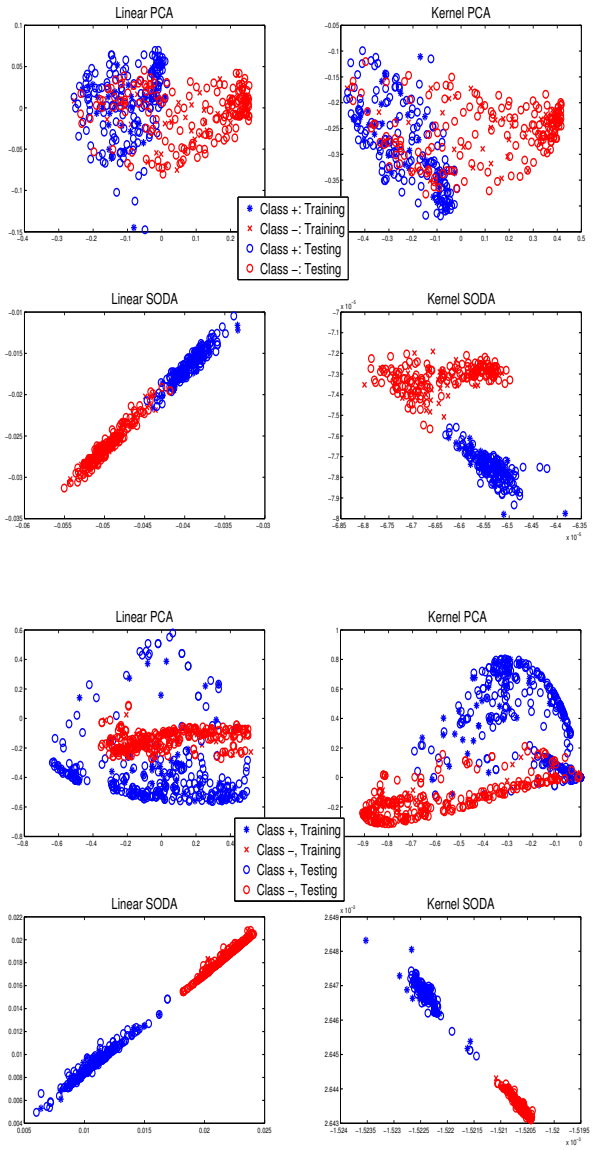

Figure 3. Visualization of the first two components of one example from data set vehicle and segment using different feature reduction techniques.