Nonlinear Feature Extraction based on Centroids and Kernel Functions

Cheonghee Park and Haesun Park*

Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455

December 19, 2002

Abstract

A nonlinear feature extraction method is presented which can reduce the data dimension down to the number of clusters, providing dramatic savings in computational costs. The dimension reducing nonlinear transformation is obtained by implicitly mapping the input data into a feature space using a kernel function, and then finding a linear mapping based on an orthonormal basis of centroids in the feature space that maximally separates the between-cluster relationship. The experimental results demonstrate that our method is capable of extracting nonlinear features effectively so that competitive performance of classification can be obtained with linear classifiers in the dimension reduced space.

Keywords. cluster structure, dimension reduction, kernel functions, Kernel Orthogonal Centroid (KOC) method, linear discriminant analysis, nonlinear feature extraction, pattern classification, support vector machines

1 Introduction

Dimension reduction in data analysis is an important preprocessing step for speeding up the main tasks and reducing the effect of noise. Nowadays, as the amount of data grows larger, extracting

*The correspondence should be addressed to Prof. Haesun Park (hpark@cs.umn.edu). This work was supported in part by the National Science Foundation grants CCR-9901992 and CCR-0204109. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

the right features is not only a useful preprocess step but becomes necessary for efficient and effective processing, especially for high dimensional data. The Principal Component Analysis (PCA) and the Linear Discriminant Analysis (LDA) are two of the most commonly used dimension reduction methods. These methods search optimal directions for the projection of input data onto a lower dimensional space [1, 2, 3]. While the PCA finds the direction along which the data scatterness is greatest, the LDA searches the direction which maximizes the between-cluster scatter and minimizes the within-cluster scatter. However, these methods have a limitation for the data which are not linearly separable since it is difficult to capture a nonlinear relationship with a linear mapping. In order to overcome such a limitation, nonlinear extensions of these methods have been proposed [4, 5, 6].

One way for a nonlinear extension is to lift the input space to a higher dimensional feature space by a nonlinear feature mapping and then to find a linear dimension reduction in the feature space. It is well known that kernel functions allow such nonlinear extensions without explicitly forming a nonlinear mapping or a feature space, as long as the problem formulation involves only the *inner products* between the data points and never the data points themselves [7, 8, 9]. The remarkable success of the support vector machine learning is an example of the effective use of the kernel functions [10, 11, 12, 13]. The kernel Principal Component Analysis (kernel PCA) [14] and the generalized Discriminant Analysis [15, 16, 17, 18] have recently been introduced as nonlinear generalizations of the PCA and the LDA by kernel functions, respectively, and some interesting experimental results are presented. However, the PCA and the LDA require solutions from the singular value decomposition and generalized eigenvalue problem, respectively. In general, these decompositions are expensive to compute when the training data set is large and especially when the problem dimension becomes higher due to the mapping to a feature space. In addition, the dimension reduction from the PCA does not reflect the clustered structure in the data well [19].

The centroid of each cluster minimizes the sum of the squared distances to vectors within the cluster and it yields a rank one approximation of the cluster [19]. In the Orthogonal Centroid method [19] the centroids are taken as representatives of each cluster and the vectors of the input space are transformed by an orthonormal basis of the space spanned by the centroids. This method provides a dimension reducing linear transformation preserving the clustering structure in the given data. The relationship between any data points and centroids measured by L_2-norm or cosine in the full dimensional space is completely preserved in the reduced space obtained with this transformation [19, 20]. Also it is shown that this method maximizes between-cluster scatter over all the transformations with orthonormal vectors [21, 22].

In this paper, we apply the centroid-based orthogonal transformation, the Orthogonal Centroid algorithm, to the data transformed by kernel-based nonlinear mapping and show that it can extract nonlinear features effectively, thus reducing the data dimension down to the number of clusters and saving the relative computational cost. In Section 2, we briefly review the Orthogonal Centroid method which is a dimension reduction method based on an orthonormal basis for the centroids. In Section 3, we derive the new Kernel Orthogonal Centroid method extending the Orthogonal Centroid method using kernel functions to handle nonlinear feature extraction and an-

alyze the computational complexity of our new method. Our experimental results presented in Section 4 demonstrate that the new nonlinear Orthogonal Centroid method is capable of extracting nonlinear features effectively so that competitive classification performance can be obtained with linear classifiers after nonlinear dimension reduction. In addition, it is shown that once we obtain a lower dimensional representation, a *linear* soft margin Support Vector Machine (SVM) is able to achieve high classification accuracy with much less number of support vectors, thus reducing prediction costs as well.

2 Orthogonal Centroid Method

Given a vector space representation,

$$A = [a_1, \cdots, a_n] \in \mathbb{R}^{m \times n}$$

of a data set of n vectors in a m-dimensional space, dimension reduction by linear transformation is to find $G^T \in \mathbb{R}^{l \times m}$ that maps a vector x to a vector $\hat{x}$ for some $l < m$:

$$G^T : x \in \mathbb{R}^{m \times 1} \rightarrow \hat{x} \in \mathbb{R}^{l \times 1} \quad i.e. \quad G^T x = \hat{x}. \tag{1}$$

In particular, we seek for a dimension reducing transformation G^T with which the cluster structure existing in the given data A is preserved in the reduced dimensional space. Eqn. (1) can be rephrased as finding a rank reducing approximation of A such that

$$\min_{G,Y} \|A - GY\|_F, \text{ where } G \in \mathbb{R}^{m \times l} \text{ and } Y \in \mathbb{R}^{l \times n}. \tag{2}$$

For simplicity, we assume that the data matrix A is clustered into r clusters as

$$A = [A_1, A_2, \cdots, A_r] \quad \text{where} \quad A_i \in \mathbb{R}^{m \times n_i}, \quad \text{and} \quad \sum_{i=1}^{r} n_i = n. \tag{3}$$

Let N_i denote the set of column indices that belong to the cluster i. The centroid c_i of each cluster A_i is the average of the columns in A_i, i.e.,

$$c_i = \frac{1}{n_i} A_i e_i \quad \text{where} \quad e_i = [1, \ldots, 1]^T \in \mathbb{R}^{n_i \times 1} \tag{4}$$

and the global centroid c is defined as

$$c = \frac{1}{n} \sum_{j=1}^{n} a_j. \tag{5}$$

Algorithm 1 Orthogonal Centroid method

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with r clusters and a data point $x \in \mathbb{R}^{m\times 1}$, it computes a matrix $Q_r \in \mathbb{R}^{m\times r}$ and gives a r-dimensional representation $\hat{x} = {Q_r}^T x \in \mathbb{R}^{r\times 1}$.

1. Compute the centroid c_i of the ith cluster for $1 \leq i \leq r$.
2. Set the centroid matrix $C = [c_1, c_2, \cdots, c_r]$.
3. Compute an orthogonal decomposition of C, which is $C = Q_r R$.
4. $\hat{x} = Q_r^T x$ gives a r-dimensional representation of x.

The centroid of each cluster is the vector that minimizes the sum of squared distances to vectors within the cluster. That is the centroid vector c_i gives the smallest distance in Frobenius norm between the matrix A_i and the rank one approximation xe_i^T where

$$\|A_i - c_i e_i^T\|_F^2 = \sum_{j\in N_i} \|a_j - c_i\|_2^2 = \min_{x\in\mathbb{R}^{m\times 1}} \sum_{j\in N_i} \|a_j - x\|_2^2 = \min_{x\in\mathbb{R}^{m\times 1}} \|A_i - xe_i^T\|_F^2. \tag{6}$$

Taking the centroids as representatives of the clusters, we find an orthonormal basis of the space spanned by the centroids by computing an orthogonal decomposition

$$C = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{7}$$

of the centroid matrix

$$C = [c_1, \cdots, c_r] \in \mathbb{R}^{m\times r},$$

where $Q = [q_1, \cdots, q_m] \in \mathbb{R}^{m\times m}$ is an orthogonal matrix with orthonormal columns and $R \in \mathbb{R}^{r\times r}$ is an upper triangular matrix. Taking the first r columns of Q, we obtain

$$C = Q_r R \quad \text{with} \quad Q_r = [q_1, \cdots, q_r], \tag{8}$$

where the columns of Q_r is an orthonormal basis for $Range(C)$ spanned by the columns of C when the columns of C are linearly independent. The algorithm can easily be modified when the columns of C are not linearly independent. The matrix ${Q_r}^T$ gives a dimension reducing linear transformation preserving the clustering structure in the sense that the relationship between any data item and a centroid measured using L_2-norm or cosine in the full dimensional space is completely preserved in the reduced dimensional space [19, 20]. This method is called the Orthogonal Centroid method and is summarized in Algorithm 1. Moreover, as we show in Section 3, this assumption is no longer needed in our new Kernel Orthogonal Centroid method.

It is shown that the linear transformation obtained in the Orthogonal Centroid method solves a trace optimization problem, providing a link between the methods of linear discriminant analysis

and those based on centroids [22]. Dimension reduction by the Linear Discriminant Analysis searches for a linear transformation which maximizes the between-cluster scatter and minimizes the within-cluster scatter. The between-cluster scatter matrix is defined as

$$S_b = \sum_{i=1}^{r} \sum_{j \in N_i} (c_i - c)(c_i - c)^T = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^T \tag{9}$$

and

$$\mathrm{trace}(S_b) = \sum_{i=1}^{r} \sum_{j \in N_i} (c_i - c)^T (c_i - c) = \sum_{i=1}^{r} \sum_{j \in N_i} \|c_i - c\|_2^2. \tag{10}$$

Let's consider a criterion that involves only the between-cluster scatter matrix, i.e., to find a dimension reducing transformation $G^T \in \mathbb{R}^{l \times m}$ such that the columns of G are orthonormal and $\mathrm{trace}(G^T S_b G)$ is maximized. Note that $\mathrm{rank}(S_b)$ can not exceed $r - 1$. Accordingly,

$$\mathrm{trace}(S_b) = \lambda_1 + \cdots + \lambda_{r-1} \tag{11}$$

where λ_i's, $1 \leq i \leq r-1$, are the $r-1$ eigenvalues of S_b. Denoting the corresponding eigenvectors as u_i's, for any $l \geq r - 1$ and $U_l = [u_1, \cdots, u_l]$, we have

$$\mathrm{trace}(U_l^T S_b U_l) = \lambda_1 + \cdots + \lambda_l = \lambda_1 + \cdots + \lambda_{r-1}. \tag{12}$$

In addition, for any $G \in \mathbb{R}^{m \times l}$ which has orthonormal columns,

$$\mathrm{trace}(G^T S_b G) \leq \mathrm{trace}(S_b). \tag{13}$$

Hence $\mathrm{trace}(G^T S_b G)$ is maximized when G is chosen as U_l for any $l \geq r - 1$ and

$$\mathrm{trace}(U_l^T S_b U_l) = \mathrm{trace}(S_b), \tag{14}$$

according to Eqns. (11) and (12).

For an eigenvalue and eigenvector pair (λ, u) of S_b, we have

$$\lambda u = S_b u = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^T u = \sum_{i=1}^{r} (n_i (c_i - c)^T u)(c_i - c). \tag{15}$$

Therefore, $u \in \mathrm{span}\{c_i - c | 1 \leq i \leq r\}$, and $u \in \mathrm{span}\{c_i | 1 \leq i \leq r\}$. Finally, the orthogonal decomposition $C = Q_r R$ of the centroid matrix $C = [c_1, c_2, \cdots, c_r]$ in the Orthogonal Centroid method gives

$$Range(Q_r) = Range(C) = Range(U_r). \tag{16}$$

Eqn. (16) implies that

$$Q_r = U_r W \tag{17}$$

for some orthogonal matrix $W \in \mathbb{R}^{r\times r}$. Since

$$\mathrm{trace}(G^T S_b G) = \mathrm{trace}(W^T G^T S_b G W)$$

for any orthogonal matrix $W \in \mathbb{R}^{r\times r}$ (see [21] for more details), Q_r also satisfies

$$\mathrm{trace}(Q_r^T S_b Q_r) = \mathrm{trace}(S_b). \tag{18}$$

So, instead of computing the eigenvectors u_i's, $i = 1, \cdots, r-1$, we simply need to compute Q_r, which is much less costly. Therefore, by computing an orthogonal decomposition of the centroid matrix we obtain a solution that maximizes $\mathrm{trace}(G^T S_b G)$ over all G with orthonormal columns.

3 Kernel Orthogonal Centroid Method

Although a linear hyperplane is a natural choice as a boundary to separate clusters it has limitations for nonlinearly structured data. To overcome this limitation we map input data to a feature space (possibly an infinite dimensional space) through a nonlinear feature mapping

$$\Phi : \mathcal{S} \subset \mathbb{R}^{m\times n} \rightarrow \mathcal{F} \subset \mathbb{R}^{N\times n} \tag{19}$$

which transforms input data into linearly separable structure. Without knowing the feature mapping Φ or the feature space $\mathcal{F}$ explicitly, we can work on the feature space $\mathcal{F}$ through kernel functions, as long as the problem formulation depends only on the inner products between data points in $\mathcal{F}$ and not on the data points themselves. For any kernel function κ satisfying Mercer's condition, there exists a reproducing kernel Hilbert space H and a feature map Φ such that

$$\kappa(x, y) =< \Phi(x), \Phi(y) > \tag{20}$$

where $<, >$ is an inner product in H [9, 23, 24]. As positive definite kernel functions satisfying Mercer's condition, polynomial kernel

$$\kappa(x, y) = (\gamma_1(x \cdot y) + \gamma_2)^d, d > 0 \text{ and } \gamma_1, \gamma_2 \in \mathbb{R} \tag{21}$$

and Gaussian kernel

$$\kappa(x, y) = \exp(-\|x - y\|^2/\sigma), \sigma \in \mathbb{R} \tag{22}$$

are in wide use.

Next we show how the Orthogonal Centroid algorithm can be combined with the kernel function to produce a nonlinear dimension reduction method which does not require the feature mapping Φ or the feature space $\mathcal{F}$ explicitly. Let Φ be a feature mapping and $\mathcal{C}$ be the centroid matrix of $\Phi(A)$, where the input data matrix A has r clusters. Consider the orthogonal decomposition

$$\mathcal{C} = \mathcal{Q}_r \mathcal{R} \tag{23}$$

of $\mathcal{C}$ where $\mathcal{Q}_r \in \mathbb{R}^{N\times r}$ has orthonormal columns and $\mathcal{R} \in \mathbb{R}^{r\times r}$ is a nonsingular upper triangular matrix [25]. We apply the Orthogonal Centroid algorithm to $\Phi(A)$ to reduce the data dimension to r, the number of clusters in the input data. Then for any data point $x \in \mathbb{R}^{m\times 1}$, the dimension reduced representation of x in a r-dimensional space will be given by $\mathcal{Q}_r^T \Phi(x)$.

We now show how we can calculate $\mathcal{Q}_r^T \Phi(x)$ without knowing Φ explicitly, i.e., without knowing $\mathcal{C}$ explicitly. The centroid matrix $\mathcal{C}$ in the feature space is

$$\mathcal{C} = \left[\frac{1}{n_1} \sum_{i\in N_1} \Phi(a_i) \, , \cdots , \frac{1}{n_r} \sum_{i\in N_r} \Phi(a_i) \right] \in \mathbb{R}^{N\times r}. \tag{24}$$

Hence

$$\mathcal{C}^T \mathcal{C} = M^T K M, \tag{25}$$

where $K \in \mathbb{R}^{n\times n}$ is the kernel matrix with

$$K(i,j) = \kappa(a_i, a_j) = < \Phi(a_i), \Phi(a_j) > \text{ for } 1 \le i,j \le n \tag{26}$$

and

$$M^T = \begin{bmatrix} \frac{1}{n_1} & \cdots & \frac{1}{n_1} & 0 & & & & \cdots & & & & 0 \\ 0 & \cdots & 0 & \frac{1}{n_2} & \cdots & \frac{1}{n_2} & 0 & \cdots & & & \cdots & 0 \\ & & & & & & \ddots & & & & & \\ 0 & & & & \cdots & & & 0 & \frac{1}{n_r} & \cdots & \frac{1}{n_r} & \end{bmatrix} \in \mathbb{R}^{r\times n}. \tag{27}$$

Since the kernel matrix K is symmetric positive definite and the matrix M has linearly independent columns, $\mathcal{C}^T\mathcal{C}$ is also symmetric positive definite. The Cholesky decomposition of $\mathcal{C}^T\mathcal{C}$ gives a nonsingular upper triangular matrix $\mathcal{R}$ such that

$$\mathcal{C}^T \mathcal{C} = \mathcal{R}^T \mathcal{R}. \tag{28}$$

Since

$$\mathcal{Q}_r = \mathcal{C}\mathcal{R}^{-1} \tag{29}$$

Algorithm 2 Kernel Orthogonal Centroid Method

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with r clusters and index sets N_i, $i = 1, \cdots, r$ which denote the set of the column indices of the data in the cluster i, and a kernel function κ, this algorithm computes nonlinear dimension reduced representation $\hat{x} = \mathcal{Q}_r^T \Phi(x) \in \mathbb{R}^{r\times 1}$ for any input vector $x \in \mathbb{R}^{m\times 1}$.

1. Formulate the kernel matrix K based on the kernel function κ as

$$K(i,j) = \kappa(a_i, a_j), 1 \le i, j \le n.$$

2. Compute $\mathcal{C}^T\mathcal{C} = M^T K M$ where

$$M(i,j) = \begin{cases} 1/n_j & \text{if } i \in N_j \\ 0 & \text{otherwise} \end{cases}$$

3. Compute the Cholesky factor $\mathcal{R}$ of $\mathcal{C}^T\mathcal{C}$: $\mathcal{C}^T\mathcal{C} = \mathcal{R}^T\mathcal{R}$.

4. The solution $\hat{x}$ for the linear system

$$\mathcal{R}^T\hat{x} = \begin{bmatrix} \frac{1}{n_1}\sum_{i\in N_1}\kappa(a_i,x) \\ \vdots \\ \frac{1}{n_r}\sum_{i\in N_r}\kappa(a_i,x) \end{bmatrix}$$

gives r-dimensional representation of x.

from (23), we have

$$\mathcal{Q}_r^T\Phi(x) = (\mathcal{R}^{-1})^T\mathcal{C}^T\Phi(x) = (\mathcal{R}^{-1})^T \begin{bmatrix} \frac{1}{n_1}\sum_{i\in N_1}\kappa(a_i,x) \\ \vdots \\ \frac{1}{n_r}\sum_{i\in N_r}\kappa(a_i,x) \end{bmatrix}. \tag{30}$$

Due to the assumption that the kernel function κ is symmetric positive definite, the matrix $\mathcal{C}^T\mathcal{C}$ is symmetric positive definite and accordingly the centroid matrix $\mathcal{C}$ has linearly independent columns. We summarize our algorithm in Algorithm 2 the Kernel Orthogonal Centroid (KOC) method.

We now briefly discuss the computational complexity of the KOC algorithm where one flop (floating point operation) represents roughly what is required to do one addition (or subtraction) and one multiplication (or division) [26]. We did not include the cost for evaluating the kernel functions $\kappa(a_i, a_j)$ and $\kappa(a_i, x)$ since this is required in any kernel-based methods, and the cost

depends on the specific kernel function. In Algorithm 2, the computation of

$$\mathcal{C}^T\mathcal{C} = M^T K M$$

requires $n^2 + rn$ flops taking advantage of the special structure of the matrix M. Cholesky decomposition of $\mathcal{C}^T\mathcal{C}$ for obtaining the upper triangular matrix $\mathcal{R}$ in (28) takes $O(\frac{r^3}{6})$ flops since $\mathcal{C}^T\mathcal{C}$ is $r \times r$ where r is the number of clusters. Once we obtain the upper triangular matrix $\mathcal{R}$, then the lower dimensional representation $\hat{x} = \mathcal{Q}_r^T\Phi(x)$ of a specific input x can be computed without computing $\mathcal{R}^{-1}$, but from solving a linear system

$$\mathcal{R}^T\hat{x} = \begin{bmatrix} \frac{1}{n_1}\sum_{i\in N_1}\kappa(a_i, x) \\ \vdots \\ \frac{1}{n_r}\sum_{i\in N_r}\kappa(a_i, x) \end{bmatrix}, \tag{31}$$

which requires $O(\frac{r^2}{2} + n)$ flops. Typically the number of clusters is much smaller than the total number of training samples n. Therefore, the complexity in nonlinear dimensional reduction by the Kernel Orthogonal Centroid method presented in Algorithm 2 is $O(n^2)$. However, the kernel-based LDA or PCA needs to handle an eigenvalue problem of size $n \times n$ where n is the number of training samples, which is more expensive to compute [14, 15, 16]. Therefore, the Kernel Orthogonal Centroid method is an efficient dimension reduction method that can reflect the nonlinear cluster relation in the reduced dimensional representation.

Alternatively, the dimension reduced representation $\mathcal{Q}_r^T\Phi(x)$ given by the KOC method can be derived as follows. Represent the centroid matrix $\mathcal{C}$ in the feature space, given by Eqn. (24), as

$$\mathcal{C} = [\tilde{c}_1, \cdots, \tilde{c}_r] = \left[\sum_{i=1}^{n}\beta_i^1\Phi(a_i), \cdots, \sum_{i=1}^{n}\beta_i^r\Phi(a_i)\right] \tag{32}$$

where β_i^j is $\frac{1}{n_j}$ if a_i belongs to the cluster j, otherwise β_i^j is 0. Now, consider the orthogonal decomposition

$$\mathcal{C} = \mathcal{Q}_r\mathcal{R} \tag{33}$$

of $\mathcal{C}$. Since the columns of $\mathcal{Q}_r$ can be represented as a linear combination of the columns of $\mathcal{C}$, they in turn can be expressed as a linear combination of the vectors $\Phi(a_i)$, $i = 1, \cdots, n$, as

$$\mathcal{Q}_r = [\tilde{q}_1, \cdots, \tilde{q}_r] = \left[\sum_{i=1}^{n}\alpha_i^1\Phi(a_i), \cdots, \sum_{i=1}^{n}\alpha_i^r\Phi(a_i)\right]. \tag{34}$$

In order to compute $\mathcal{Q}_r^T\Phi(x)$, first we will show how we can find the coefficients α_i^j's from the

given β_i^j's where

$$\begin{aligned} \mathcal{C} &= \left[\sum_{i=1}^{n} \beta_i^1 \Phi(a_i), \cdots, \sum_{i=1}^{n} \beta_i^r \Phi(a_i)\right] \\ &= \left[\sum_{i=1}^{n} \alpha_i^1 \Phi(a_i), \cdots, \sum_{i=1}^{n} \alpha_i^r \Phi(a_i)\right] \begin{bmatrix} \tilde{r}_{11} & \cdots & \tilde{r}_{1r} \\ 0 & \ddots & \vdots \\ 0 & 0 & \tilde{r}_{rr} \end{bmatrix} \end{aligned} \tag{35}$$

without knowing Φ explicitly. Note that we can calculate inner products between the centroids in the feature space through the kernel matrix K as

$$< \tilde{c}_s, \tilde{c}_t >= \left(\sum_{i=1}^{n} \beta_i^s \Phi(a_i)\right)^T \left(\sum_{i=1}^{n} \beta_i^t \Phi(a_i)\right) = (\beta^s)^T K \beta^t, \tag{36}$$

where $\beta^s = [\beta_1^s, \cdots, \beta_n^s]^T$. In addition, the vectors

$$\tilde{p}_s = \frac{\tilde{c}_s}{\sqrt{< \tilde{c}_s, \tilde{c}_s >}} \quad \text{and} \quad \tilde{p}_t = \tilde{c}_t - \frac{< \tilde{c}_s, \tilde{c}_t >}{< \tilde{c}_s, \tilde{c}_s >} \tilde{c}_s, \quad 1 \leq s \leq t \leq r, \tag{37}$$

that would appear in the modified Gram-Schmidt process of computing $\mathcal{Q}_r$ are orthogonal vectors such that

$$span\{\tilde{p}_s, \tilde{p}_t\} = span\{\tilde{c}_s, \tilde{c}_t\}. \tag{38}$$

From Eqns. (36) and (37), we can represent $\tilde{p}_s$ and $\tilde{p}_t$ as linear combinations of $\Phi(a_i)$, $i = 1, \cdots, n$. Based on these observations, we can apply the modified Gram-Schmidt method [25] to the centroid matrix $\mathcal{C}$ to compute an orthonormal basis of the centroids, even though we only have an implicit representation of the centroids in the feature space. Once the orthonormal basis $\mathcal{Q}_r$ is obtained, i.e., the coefficients α_i^s's of $\tilde{q}_s = \sum_{i=1}^{n} \alpha_i^s \Phi(a_i)$, $1 \leq s \leq r$ are found, then the reduced dimensional representation $\mathcal{Q}_r^T \Phi(x)$ can be computed from

$$\mathcal{Q}_r^T \Phi(x) = \begin{bmatrix} \sum_{i=1}^{n} \alpha_i^1 \kappa(a_i, x) \\ \vdots \\ \sum_{i=1}^{n} \alpha_i^r \kappa(a_i, x) \end{bmatrix}. \tag{39}$$

This approach is summarized in Algorithm 3.

Algorithm 3, the Kernel Orthogonal Centroid method, requires $O(\frac{r^2}{2} n^2)$ flops for the orthogonal decomposition of the centroid matrix $\mathcal{C}$ and $O(rn)$ flops for obtaining the reduced dimensional representation $\mathcal{Q}_r^T \Phi(x)$ for any input vector $x \in \mathbb{R}^{m \times 1}$. Hence the total complexity of Algorithm 3 is slightly higher than Algorithm 2. However, the approach of finding the parameters α^s

Algorithm 3 Kernel Orthogonal Centroid method by the modified Gram-Schmidt

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with r clusters and a kernel function κ, this method computes the nonlinear dimension reduced representation $\hat{x} = \mathcal{Q}_r^T \Phi(x) \in \mathbb{R}^{r\times 1}$ for any input vector $x \in \mathbb{R}^{m\times 1}$.

1. Define $\beta_i^j = \begin{cases} \frac{1}{n_j} & \text{if } a_i \text{ belongs to the cluster } j \\ 0 & \text{otherwise} \end{cases}$ for $1 \le i \le n, 1 \le j \le r$.

2. Compute an orthogonal decomposition $\mathcal{C} = \mathcal{Q}_r \mathcal{R}$ of the centroid matrix $\mathcal{C}$ as in Eqn. (35) by the modified Gram-Schmidt.

   for $s = 1, \cdots, r$
   $\tilde{r}_{ss} = \sqrt{< \tilde{c}_s, \tilde{c}_s >} = \sqrt{(\beta^s)^T K \beta^s}$
   $\alpha^s = \beta^s / \tilde{r}_{ss}$
   for $t = s+1, \cdots, r$
   $\tilde{r}_{st} =< \tilde{q}_s, \tilde{c}_t >= (\alpha^s)^T K \beta^t$
   $\beta^t = \beta^t - \alpha^s \tilde{r}_{st}$
   end
   end

3. $\mathcal{Q}_r^T \Phi(x) = [\, \sum_{i=1}^n \alpha_i^1 \kappa(a_i, x), \cdots, \sum_{i=1}^n \alpha_i^r \kappa(a_i, x) \,]^T$.

from the parameters β^s can be applied in other context of kernel based feature extraction where direct derivation of the kernel based method as in Algorithm 2 is not possible. We have applied a similar approach in developing nonlinear discriminant analysis based on the generalized singular value decomposition, which works successfully regardless of nonsingularity of the within-cluster scatter matrix [27]. More discussions about the optimization criteria used in LDA, including the within-cluster scatter matrix, are given in the next section.

In the Kernel Orthogonal Centroid method, the choice of kernel function will influence the results as in any other kernel-based methods. However, a general guideline for an optimal choice of the kernel is difficult to obtain. In the next section, we present the numerical test results that compare the effectiveness of our proposed method to other existing methods. We also visualize the effects of various kernels in our algorithms.

4 Computational Test Results

The Kernel Orthogonal Centroid method has been implemented in C on IBM SP at the University of Minnesota Supercomputing Institute in order to investigate its computational performance. The prediction accuracy of classification of the test data whose dimension was reduced to the number of clusters by our KOC method was compared to other existing linear and nonlinear feature extraction

methods. We used data sets available in the public domain as well as some artificial data we generated. In addition, the input data with cluster structure are visualized in the 3-dimensional space after dimension reduction by our proposed method to illustrate the quality of the represented clustered structure. In the process, we also illustrate the effect of various kernel functions. We used two of the most commonly used kernels in our KOC method, which are polynomial kernels

$$\kappa(x, y) = (x \cdot y + 1)^d, d > 0$$

and Gaussian kernels

$$\kappa(x, y) = \exp(-\|x - y\|^2/\sigma), \sigma \in \mathbb{R}.$$

Our experimental results illustrate that when the Orthogonal Centroid method is combined with a nonlinear mapping, as in the KOC algorithm, with an appropriate kernel function, the linear separability of the data is increased in the reduced dimensional space. This is due to the nonlinear dimension reduction achieved by the orthonormal basis of the centroids in the feature space which maximizes the between-cluster scatter.

4.1 3D Representation of Nonseparable Data

The purpose of our first test is to illustrate how our method produces a lower dimensional representation separating the data items which belong to different classes. We present the results from the Iris plants data of Fisher [28], as well as from an artificial data set that we generated, where the data points in three clusters in the original space are not separable.

In the Iris data, the given data set has 150 data points in a 4-dimensional space and is clustered to 3 classes. One class is linearly separable from the other two classes, but the latter two classes are not linearly separable. Figure 1 shows the data points which are reduced to a 3-dimensional space by various dimension reduction methods. The leftmost figure in Figure 1 is obtained by an optimal rank 3 approximation of the data set from its singular value decomposition, which is one of the most commonly used techniques for dimension reduction [25]. The figure shows that after the dimension reduction by a rank 3 approximation from the SVD, two of the three classes are still not quite separated. The second and the third figures in Figure 1 are obtained by our KOC method with the Gaussian kernel where $\sigma = 1$ and 0.01, respectively. They show that our Kernel Orthogonal Centroid method combined with the Gaussian kernel function with $\sigma = 0.01$ gives a 3-dimensional representation of Iris data where all three clusters are well separated and the between-cluster relationship is remote.

The artificial data we generated has three classes. Each class consists of 200 data points uniformly distributed in the cubic region with height 1.4, width 4 and length 18.5. The three classes intersect each other as shown in the top left figure of Figure 2, for the total of 600 given data points. Different kernel functions were applied to obtain the nonlinear representation of these given data points. In this test, the dimension of the original data set is in fact not reduced, since it

was given in the 3-dimensional space, and after applying the KOC method, the final dimension is also 3 which is the number of the clusters. The right top figure shows the new data representation with a polynomial kernel of degree 4. The lower figures are produced using the Gaussian kernel $\kappa(x, y) = \exp(-\|x-y\|^2/\sigma)$ where $\sigma = 5$ (the left figure) and 0.05 (the right figure), respectively. As with the Iris data, with the proper kernel function, the three clusters are well separated. It is interesting to note that the within-cluster relationship also became tighter although the dimension reduction criterion involves only the between-cluster relationship.

4.2 Performance in Classification

In our second test, the purpose was to compare the effectiveness of dimension reduction from our KOC method in classification. For this purpose, we compared the accuracy of binary classification results where the dimension of the data items are reduced by our KOC method as well as by the kernel Fisher discriminant (KFD) method of Mika et al. [15]. The test results presented in this section are for binary classifications for comparisons to KFD which can handle two-class cases only. For more details on the test data generation and results, see [15], where the authors presented the kernel Fisher Discriminant(KFD) method for the binary-class with substantial test results comparing their method to other classifiers.

The Linear Discriminant Analysis optimizes various criteria functions which involve between-cluster, within-cluster or mixed-cluster scatter matrices [2]. Many of the commonly used criteria involve the inverse of the within-cluster scatter matrix S_w, which is defined as,

$$S_w = \sum_{i=1}^{r} \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^T, \tag{40}$$

requiring this within-cluster scatter matrix S_w to be nonsingular. However, in many applications the matrix S_w is either singular or ill-conditioned. One common situation when S_w becomes singular is when the number of data points is smaller than the dimension of the space where each data item resides. Numerous methods have been proposed to overcome this difficulty including the regularization method [29]. A method Howland et al. recently developed called LDA/GSVD, which is based on the generalized singular value decomposition, works well regardless of the singularity of the within-cluster scatter matrix. (See [21].) In the KFD analysis, Mika et al. used regularization parameters to make the within-cluster scatter matrix nonsingular.

Fisher discriminant criterion requires a solution of an eigenvalue problem which is expensive to compute. In order to improve the computational efficiency of KFD, several methods have been proposed, which include the KFD based on a quadratic optimization problem using regularization operators or a sparse greedy approximation [30, 31, 32]. In general, quadratic optimization problems are as costly as the eigenvalue problems. A major advantage of our KOC method is that its computational cost is substantially lower, requiring computation of a Cholesky factor and a solution for a linear system where the problem size is the same as the number of clusters. The

computational savings come from the fact that the within-cluster scatter matrix is not involved in the optimal dimension reduction criterion [22].

In Table 1, we present the implementation results on seven data sets which Mika et al. have used in their tests[1] [33]. The data sets which are not already clustered or with more than two clusters were reorganized so that the results have only two classes. Each data set has 100 pairs of training and test data items which were generated from one pool of data items. For each data set, the average accuracy is calculated by running these 100 cases. Parameters for the best candidate for the kernel function and SVM are determined based on a 5 fold cross-validation using the first five training sets. We repeat their results in the first five columns of Table 1 which show the prediction accuracies in percentage (%) from the RBF classifier(RBF), AdaBoost(AB), regularized AdaBoost, SVM and KFD. For more details, see [15].

The results shown in the column for KOC are obtained from the *linear* soft margin SVM classification using the software svm^{light} [34] after dimension reduction by KOC. The test results with the polynomial kernel with degree 3 and the Gaussian kernel with an optimal σ value for each data set are presented in Table 1. The results show that our method obtained comparable accuracy to other methods in all the tests we performed. Using our KOC algorithm, we were able to achieve substantial computational savings not only due to the lower computational complexity of our algorithm, but from using a *linear* SVM. Since no kernel function (or *identity* kernel function) is involved in the classification process by a linear SVM, the parameter w in the representation of the optimal separating hyperplane

$$f(x) = w^T x + b$$

can be computed explicitly, saving substantial computation time in the testing stage. In addition, due to the dimension reduction, kernel function values are computed between much shorter vectors.

Another phenomenon we observed in all these tests is that after the dimension reduction by KOC, the linear soft margin SVM requires significantly less number of training data points as the *support vectors*, compared to the soft margin SVM with the kernel function applied to the original input data. More details can be found in the next section.

4.3 Performance of the Support Vector Machines

Using the same artificial data that we used in Section 4.1, now we compare the performance of classification on the soft-margin SVMs using the data generated from our KOC, as well as using the original data. This time, 600 more test data points are generated in addition to the 600 training data generated for the earlier test in Section 4.1. The test data are generated following the same rules as the training data, but independently from the training data.

In order to apply the SVMs for a three-class problem, we used the method where after a binary classification of C_1 vs. not C_1 ($C_1/\sim C_1$) is determined, data classified not to be in the class

[1]The breast cancer data set was obtained from the University Medical Center, Inst. of Oncology, Ljubljana, Yugoslavia. Thanks to M. Zwitter and M. Soklic for the data.

C_1 is further classified to be in C_2 or C_3 (C_2/C_3). There are three different ways to organize the binary classifiers for a three-class problem depending on which classifier $C_i/\sim C_i$, $i = 1, 2, 3$, is considered in the first step. One may run all three cases to achieve better prediction accuracy. For more details, see [35]. We present the results obtained from $C_1/\sim C_1$ and C_2/C_3, since all three ways produced comparable results in our tests.

In Figure 3, the prediction accuracy and the number of support vectors are shown when the nonlinear soft margin SVM is applied in the original dimension and the *linear* soft margin SVM is applied in the reduced dimension obtained from our KOC algorithm. In both cases, Gaussian kernels with various σ values were used. While the best prediction accuracy among various σ values is similar in both cases, it is interesting to note that the number of support vectors is much less in the case of the linear soft margin SVM with data in the reduced space. In addition, the performance and the number of support vectors are less sensitive to the value of σ after dimension reduction by the KOC algorithm.

The test results confirm that the KOC algorithm is an effective method in extracting important nonlinear features. Once the best features are extracted, the computation of finding the optimal separating hyperplane and classification of new data become much more efficient. An added benefit we observed in all our tests is that after the kernel-based nonlinear feature extraction by the KOC algorithm, another use of the kernel function in the SVM is not necessary. Hence the simple linear SVM can be effectively used, achieving further efficiency in computation. Another merit of the KOC method is that after its dramatic dimension reduction, in the classification stage the comparison between the vectors by any similarity measure such as Euclidean distance (L_2 norm) or cosine becomes much more efficient, since we now compare the vectors with r components each, rather than m components each.

5 Conclusion

We have presented a new method for nonlinear feature extraction called the Kernel Orthogonal Centroid (KOC). The KOC method reduces the dimension of the input data down to the number of clusters. The dimension reducing nonlinear transformation is a composite of two mappings; the first implicitly maps the data into a feature space by using a kernel function, and the second mapping with orthonormal vectors in the feature space is found so that the data items belonging to different clusters are maximally separated. One of the major advantages of our KOC method is its computational efficiency, compared to other kernel-based methods such as kernel PCA [14] or KFD [15, 30, 32] and GDA [16]. The efficiency compared to other nonlinear feature extraction method utilizing discriminant analysis is achieved by only considering the between-cluster scatter relationship and by developing an algorithm which achieves this purpose from finding an orthonormal basis of the centroids, which is far cheaper than computing the eigenvectors.

The experimental results illustrate that the KOC algorithm achieves an effective lower dimensional representation of the input data which are not linearly separable, when combined with the

right kernel function. With the proposed feature extraction method, we were able to achieve comparable or better prediction accuracy to other existing classification methods in our tests. In addition, when it is used with the SVM, in all our tests the linear SVM performed as well and with far less number of support vectors, further reducing the computational costs in the test stage.

Acknowledgements

The authors would like to thank the University of Minnesota Supercomputing Institute (MSI) for providing the computing facilities. We also would like to thank Dr. S. Mika for valuable information.

References

[1] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-interscience, New York, 2001.

[2] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.

[3] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[4] M.A. Kramer. Nonlinear principal component analysis using autoassociative neural network. *AIChE journal*, 37(2):233–243, 1991.

[5] K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks: Theory and Applications*. Wiley-interscience, New York, 1996.

[6] T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American statistical association*, 89:1255–1270, 1994.

[7] M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[8] S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988.

[9] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola. Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5):1000–1017, September 1999.

[10] B.E. Boser, I.M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth annual workshop on computational learning theory, Pittsburgh, ACM*, 1992.

[11] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[12] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[13] B. Schölkopf, C. Burges, and A. Smola. *Advances in Kernel Method-Support Vector Learning*. MIT Press, 1999.

[14] B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10:1299–1319, 1998.

[15] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In E.Wilson J.Larsen and S.Douglas, editors, *Neural networks for signal processing IX*, pages 41–48. IEEE, 1999.

[16] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12:2385–2404, 2000.

[17] V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. *Advances in neural information processing systems*, 12:568–574, 2000.

[18] S.A. Billings and K.L. Lee. Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural networks*, 15(2):263–270, 2002.

[19] H. Park, M. Jeon, and J.B. Rosen. Lower dimensional representation of text data based on centroids and least squares, 2001. submitted to *BIT*.

[20] M. Jeon, H. Park, and J. B. Rosen. Dimensional reduction based on centroids and least squares for efficient processing of text data. In *Proceedings for the first SIAM international workshop on text mining, Chiago, IL*, 2001.

[21] P. Howland, M. Jeon, and H. Park. Cluster structure preserving dimension reduction based on the generalized singular value decomposition. SIAM Journal on Matrix Analysis and Applications, to appear.

[22] P. Howland and H. Park. Cluster-preserving dimension reduction methods for efficient classification of text data, a comprehensive survey of text mining. Springer-Verlag, to appear, 2002.

[23] N. Cristianini, and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge, 2000.

[24] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[25] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[26] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, first edition, 1983.

[27] C. Park and H. Park. Kernel discriminant analysis based on the generalized singular value decomposition. In preparation.

[28] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annual eugenics*, 7, Part II:179–188, 1936.

[29] J.H. Frieman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.

[30] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A.J. Smola, and K.-R. Müller. Invariant feature extraction and classification in kernel spaces. *Advances in neural information processing systems*, 12:526–532, 2000.

[31] S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the kernel fisher algorithm. *Advances in neural information processing systems*, 13, 2001.

[32] S. Mika, A.J. Smola, and B. Schölkopf. An improved training algorithm for kernel fisher discriminants. In *proceedings AISTATS, Morgan Kaufmann*, pages 98–104, 2001.

[33] http://www.first.gmd.de/∼raetsch.

[34] T.Joachims. Making large-scale SVM learning practical. LS8-Report 24, Universität Dortmund, LS VIII-Report, 1998.

[35] H. Kim and H. Park. Protein secondary structure prediction by support vector machines and position-specific scoring matrices. Submitted for publication, Oct., 2002.

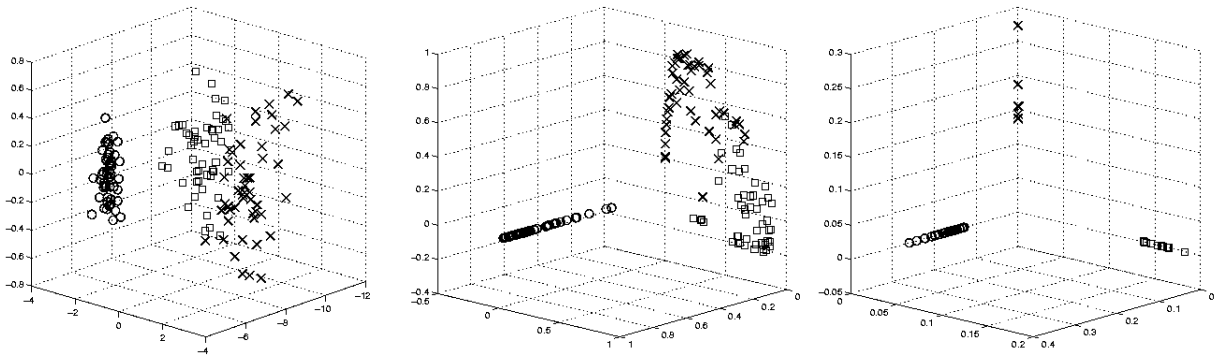

Figure 1: Iris data represented in a 3-dimensional space. The first figure is obtained from a rank 3 approximation by the SVD. The others are by the Kernel Orthogonal Centroid method with Gaussian kernel $\kappa(x, y) = \exp(-\|x - y\|^2/\sigma)$ where $\sigma = 1(\text{the second})$ and $0.01(\text{the third})$. Using Gaussian kernel with $\sigma = 0.01$, our method obtained a complete separation of the three classes.

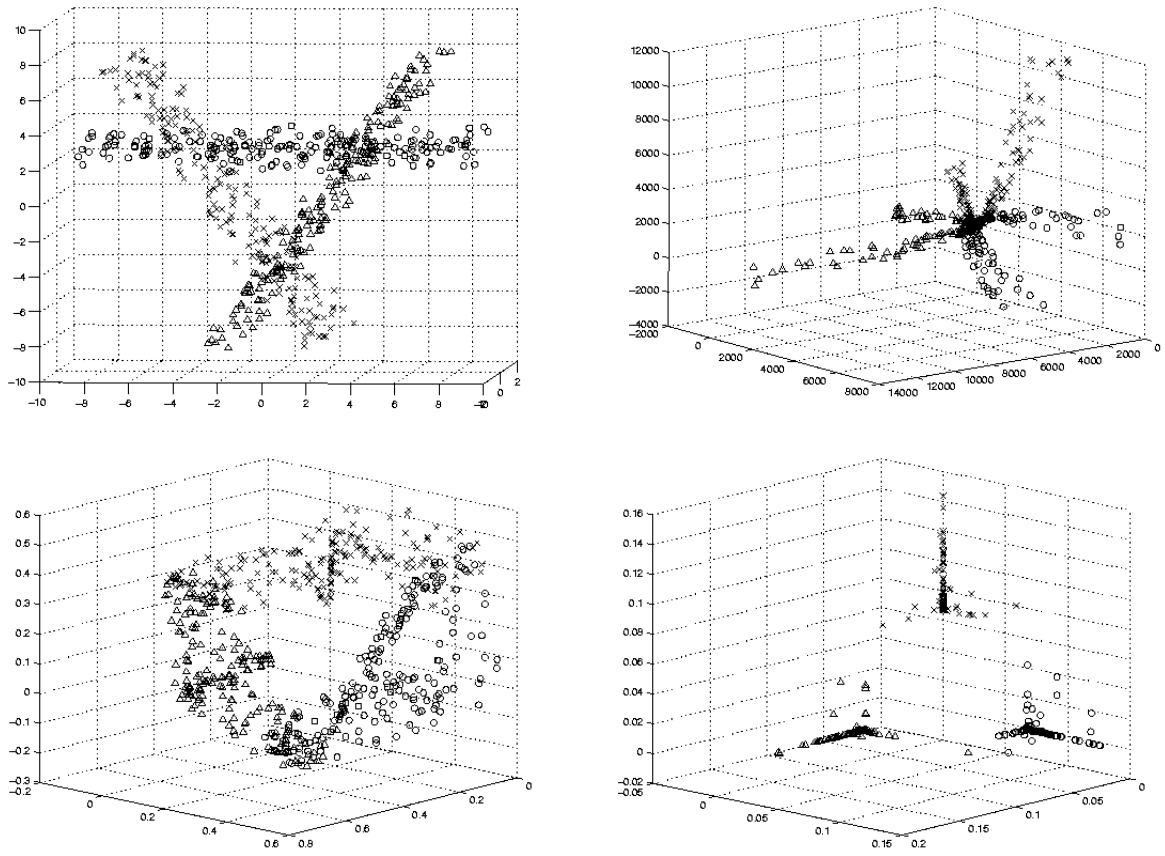

Figure 2: The top left figure is the training data with 3 clusters in a 3-dimensional space. The top right figure is generated by the Kernel Orthogonal Centroid method with a polynomial kernel of degree 4. The bottom left and bottom right figures are from the KOC algorithm using Gaussian kernels with width 5 and 0.05, respectively.

	Results from [15]					KOC		
	RBF	AB	AB_R	SVM	KFD	Gaussian		poly. d=3
Banana	**89.2**	87.7	89.1	88.5	**89.2**	**89.1**	$\sigma = 0.1$	65.9
B.cancer	72.4	69.6	73.5	74.0	**74.2**	75.0	5.0	**76.4**
German	75.3	72.5	75.7	**76.4**	76.3	**76.3**	6.0	74.6
Heart	82.4	79.7	83.5	**84.0**	83.9	83.9	49.0	**84.9**
Thyroid	95.5	95.6	95.4	95.2	**95.8**	**95.5**	1.8	88.9
Titanic	76.7	77.4	77.4	**77.6**	76.8	**77.3**	33.0	76.2
Twonorm	97.1	97.0	97.3	97.0	**97.4**	**97.6**	38.0	**97.6**

Table 1: The prediction accuracies are shown. The first part (RBF to KFD) is from [15]: classification accuracy from a single RBF classifier(RBF), AdaBoost(AB), regularized AdaBoost, SVM and KFD. The last two columns are from the Kernel Orthogonal Centroid method using Gaussian kernels (optimal σ values shown) and a polynomial kernel of degree 3. For each test, the best prediction accuracy result is shown in boldface.

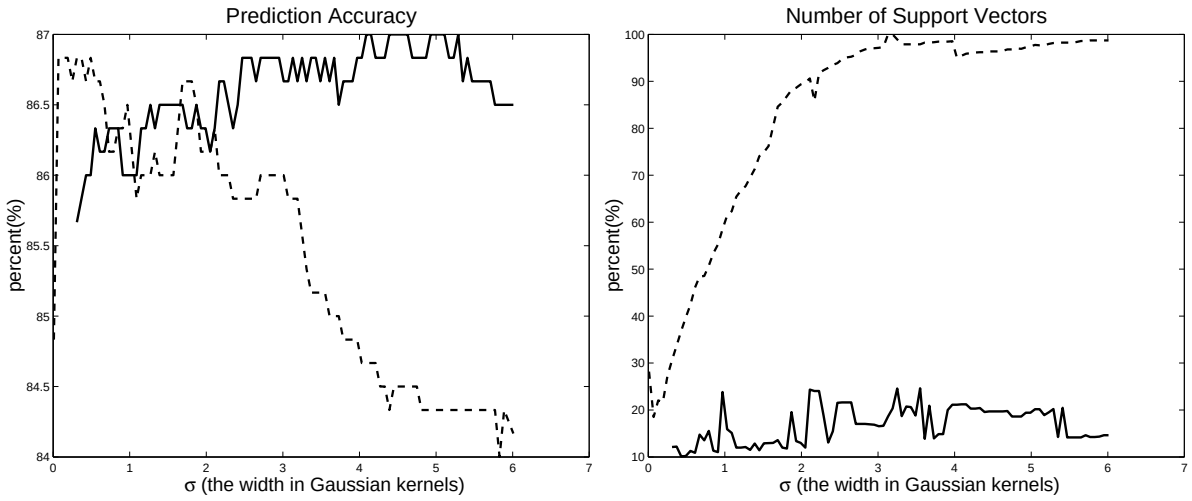

Figure 3: Classification results on the artificial data using a soft margin SVM. The left graph shows the prediction accuracy in the full input space by a SVM with a Gaussian kernel (dashed line), and that in the reduced dimensional space obtained by our KOC method with a Gaussian kernel and a linear SVM (solid line). The right graph compares the number of support vectors generated in the training process. While the best accuracy is similar in both cases, the overall number of support vectors is much less when the reduced dimensional representation is used in a linear SVM.