Self-Supervised Learning Based on Discriminative Nonlinear Features for Image Classification

Qi Tian[*1], Ying Wu[2], Jie Yu[1], Thomas S. Huang[3]

[1]Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249

[2]Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208

[3]Beckman Institute, University of Illinois, 405 N. Mathews Ave, Urbana, IL 61801

Summary

It is often tedious and expensive to label large training datasets for learning-based image classification. This problem can be alleviated by self-supervised learning techniques, which take a hybrid of labeled and unlabeled data to train classifiers. However, the feature dimension is usually very high (typically from tens to several hundreds). The learning is afflicted by the curse of dimensionality as the search space grows exponentially with the dimension. Discriminant-EM (DEM) proposed a framework for such tasks by applying self-supervised learning in an optimal discriminating subspace of the original feature space. However, the algorithm is limited by its linear transformation structure which cannot capture the non-linearity in the class distribution. This paper extends the linear DEM to a nonlinear kernel algorithm, Kernel DEM (KDEM) based on kernel multiple discriminant analysis (KMDA). KMDA provides better ability to simplify the probabilistic structures of data distribution in a discriminating subspace. KMDA and KDEM are evaluated on both benchmark databases and synthetic data. Experimental results show that classifier using KMDA is comparable with support vector machine (SVM) on standard benchmark test, and KDEM outperforms a variety of supervised and semi-supervised learning algorithms for several tasks of image classification. Extensive results show the effectiveness of our approach.

[*] Corresponding author: Qi Tian, 6900 North Loop 1604 West, Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249. Email: qitian@cs.utsa.edu Tel: (210) 458-5165, Fax: (210) 458-4437

Revised draft submitted to *Pattern Recognition*, *Special Issue on Image Understanding for Digital Photographs* June 2004

Self-Supervised Learning Based on Discriminative Nonlinear Features for Image Classification

Qi Tian[*1], Ying Wu[2], Jie Yu[1], Thomas S. Huang[3]

[1]Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249

[2]Department of Electrical and Computer Engineering, Northwestern University, IL 60208

[3]Beckman Institute, University of Illinois, 405 N. Mathews Ave., Urbana, IL 61801

Abstract

For learning-based tasks such as image classification, the feature dimension is usually very high. The learning is afflicted by the curse of dimensionality as the search space grows exponentially with the dimension. Discriminant-EM (DEM) proposed a framework by applying self-supervised learning in a discriminating subspace. This paper extends the linear DEM to a nonlinear kernel algorithm, Kernel DEM (KDEM), and evaluates KDEM extensively on benchmark image databases and synthetic data. Various comparisons with other state-of-the-art learning techniques are investigated for several tasks of image classification. Extensive results show the effectiveness of our approach.

Keywords

Discriminant analysis, kernel function, unlabeled data, support vector machine, image classification

[*] Corresponding author: Qi Tian, 6900 North Loop 1604 West, Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249. Email: qitian@cs.utsa.edu Tel: (210) 458-5165, Fax: (210) 458-4437.

1. INTRODUCTION

Content-based image retrieval (CBIR) is a technique which uses visual content to search images from large-scale image database according to userís interests and has been an active and fast advancing research area since the 1990s [1]. A challenge in content-based image retrieval is the semantic gap between the high-level semantics in human mind and the low-level features computed by the machine. Users seek semantic similarity, but the machine can only provide similarity by data processing. The mapping between them would be nonlinear such that it is impractical to represent it explicitly. A promising approach to this problem is machine learning, by which the mapping could be learned through a set of examples. The task of image retrieval is to find as many as possible ìsimilarî images to the query images in a given database. The retrieval system acts as a classifier to divide the images in the database into two classes, either relevant or irrelevant. Many supervised or semi-supervised learning approaches have been employed to approach this classification problem. Successful examples of learning approaches in content-based image retrieval can be found in the literature [2-9].

However, representing image in the image space is formidable, since the dimensionality of the image space is intractable. Dimension reduction could be achieved by identifying invariant image features. In some cases, domain knowledge could be exploited to extract image features from visual inputs. On the other hand, the generalization abilities of many current methods largely depend on training datasets. In general, good generalization requires large and representative labeled training datasets.

In content-based image retrieval [1-9], there are a limited number of labeled training images given by user query and relevance feedback [10]. Pure supervised learning from such a small training dataset will have poor generalization performance. If the learning classifier is over-trained on the small training dataset, *over-fitting* will probably occur. However, there are a large number of unlabeled images or unlabeled data in general in the given database. Unlabeled data contain information about the joint distribution over features which can be used to help supervised learning. These algorithms normally assume that only a fraction of the data is labeled with ground truth, but still take advantage of the entire data set to generate good classifiers; they make the assumption that nearby data are likely to be generated by the same class. This learning paradigm could be looked as an integration of pure supervised and unsupervised learning.

Discriminant-EM (DEM) [11] is a self-supervised learning algorithm for such purposes by taking a small set of labeled data with a large set of unlabeled data. The basic idea is to learn discriminating features and the classifier simultaneously by inserting a multi-class linear discriminant step in the standard expectation-maximization (EM) [12] iteration loop. DEM makes assumption that the probabilistic structure of data distribution in the lower dimensional discriminating space is simplified and could be captured by lower order Gaussian mixture.

Fisher discriminant analysis (FDA) and multiple discriminant analysis (MDA) [12] are traditional two-class and multi-class discriminant analysis which treats every class equally when finding the optimal projection subspaces. Contrary to FDA and MDA, Zhou and Huang [13] proposed a biased discriminant analysis (BDA) which treats all positive, i.e., relevant examples as one class, and negative, i.e., irrelevant, examples as different classes for content-based image retrieval. The intuition behind BDA is that ìall positive examples are alike, each negative example is negative in its own wayî. Compared with the state-of-the-art methods such as support vector machines (SVM) [14], BDA [13] outperforms SVM when the size of negative examples is small (<20).

However, one drawback of BDA is its ignorance of unlabeled data in the learning process. Unlabeled data could improve the classification under the assumption that nearby data is to be generated by the same class [15]. In the past years there has been a growing interest in the use of unlabeled data for enhancing classification accuracy in supervised learning such as text classification [16, 17], face expression recognition [18], and image retrieval [11, 19].

DEM differs from BDA in the use of unlabeled data and the way they treat the positive and negative examples in the discrimination step. However, the discrimination step is linear in both DEM and BDA, they have difficulty handling data sets which are not linearly separable. In CBIR, image distribution is usually modeled as a mixture of Gaussians, which is highly non-linear-separable. In this paper, we generalize the DEM from linear setting to a nonlinear one. Nonlinear, kernel discriminant analysis transforms the original data space **X** to a higher dimensional kernel ì*feature spaceî* [1] F and then projects the transformed data to a lower dimensional discriminating subspace Δ such that nonlinear discriminating features could be identified and training data could be better classified in a nonlinear feature subspace.

The rest of paper is organized as follows. In Section 2, we present nonlinear discriminant analysis using kernel functions [20, 21]. In Section 3, two schemes are presented for sampling training data for efficient learning of nonlinear kernel discriminants. In Section 4, Kernel DEM is formulated and in Section 5 we apply Kernel DEM algorithm to various applications and compare with other state-of-the-art methods. Our experiments include standard benchmark testing, image classification on Corel photos and synthetic data, and view-independent hand posture recognition. Finally, conclusions and future work are given in Section 6.

2. NONLINEAR DISCRIMINANT ANALYSIS

Preliminary results of applying DEM for CBIR have been shown in [11]. In this section, we generalize the DEM from linear setting to a nonlinear one. We first map the data **x** via a nonlinear mapping ϕ into some high, or even infinite dimensional feature space F and then apply linear DEM in the feature space F. To avoid working with the mapped data explicitly (being impossible if F is of an infinite dimension), we will adopt the well-known *kernel trick* [22]. The kernel functions $k(\mathbf{x}, \mathbf{z})$ compute a dot product in a feature

[1] A term used in kernel machine literatures to denote the new space after the non-linear transform-this shall not be confused with the *feature space* concept used in content-based image retrieval to denote the space for features or descriptors extracted from the media data.

space F: $k(\mathbf{x},\mathbf{z}) = (\phi(\mathbf{x})^T \cdot \phi(\mathbf{z}))$. Formulating the algorithms in F using ϕ only in dot products, we can replace any occurrence of a dot product by the kernel function k, which amounts to performing the same *linear* algorithm as before, but *implicitly* in a kernel feature space F. Kernel principle has quickly gained attention in image classification in recent years [13, 19-23].

2.1. Linear Features and Multiple Discriminant Analysis

It is common practice to preprocess data by extracting linear and non-linear features. In many feature extraction techniques, one has a criterion assessing the quality of a single feature which ought to be optimized. Often, one has prior information available that can be used to formulate quality criteria, or probably even more common, the features are extracted for a certain purpose, e.g., for subsequently training some classifier. What one would like to obtain is a feature, which is as invariant as possible while still covering as much of the information necessary for describing the dataís properties of interest.

A classical and well-known technique that solves thus type of problem, considering only one linear feature, is the maximization of the so called *Rayleigh* coefficient [24, 12].

$$J(W) = \frac{|W^T S_1 W|}{|W^T S_2 W|} \qquad (1)$$

Here, W denotes the weight vector of a linear feature extractor (i.e., for an example $\mathbf{x}$, the feature is given by the projections $(W^T\mathbf{x})$ and S_1 and S_2 are symmetric matrices designed such that they measure the desired information and the undesired noise along the direction W. The ratio in equation (1) is maximized when one covers as much as possible of the desired information while avoiding the undesired.

If we look for discriminating directions for classification, we can choose S_B to measure the separability of class centers (between-class variance), i.e., S_1 in equation (1), and S_W to measure the within-class variance, i.e., S_2 in equation (1). In this case, we recover the well-known Fisher discriminant [25], where S_B and S_W are given by:

$$S_B = \sum_{j=1}^{C} N_j \cdot (m_j - m)(m_j - m)^T \qquad (2)$$

$$S_W = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (x_i^{(j)} - m_j)(x_i^{(j)} - m_j)^T \qquad (3)$$

we use $\{x_i^{(j)}, i = 1,\ldots,N_j\}, j = 1,\ldots,C$ ($C = 2$ for Fisher discriminant analysis) to denote the feature vectors of training samples. *C* is the number of classes, N_j is the number of the samples of the *j*th class, $x_i^{(j)}$ is the *i*th sample from the *j*th class, m_j is mean vector of the *j*th class, and *m* is grand mean of all examples.

If S_1 in equation (1) is the covariance matrix of all the samples

$$S_1 = \tfrac{1}{C} \sum_{j=1}^{C} \tfrac{1}{N_j} \sum_{i=1}^{N_j} (x_i^{(j)} - m)(x_i^{(j)} - m)^T \qquad (4)$$

and S_2 identity matrix, we recover standard principal component analysis (PCA) [26].

If S_1 is the data covariance and S_2 the noise covariance (which can be estimated analogous to equation (4), but over examples sampled from the assumed noise distribution), we obtain oriented PCA [26], which aims at finding a direction that describes most variance in the data while avoiding known noise as much as possible.

PCA and FDA, i.e., linear discriminant analysis (LDA) are both common techniques for feature dimension reduction. LDA constructs the *most discriminative* features while PCA constructs the *most descriptive* features in the sense of packing most ì energyî .

There has been a tendency to prefer LDA over PCA because, as intuition would suggest, the former deals directly with discrimination between classes, whereas the latter deals without paying particular attention to the underlying class structure. An interesting result is reported by Martinez and Kaka [27] that this is not always true in their study on face recognition. PCA might outperform LDA when the number of samples per class is small or when the training data non-uniformly sample the underlying distribution.

When the number of training samples is large and training data is representative for each class, LDA will outperform PCA.

Multiple discriminant analysis (MDA) is a natural generalization of Fisherís linear discriminative analysis for multiple classes [12]. The goal is to maximize the ratio of equation (1). The advantage of using this ratio is that it has been proven in [28] that if S_W is nonsingular matrix then this ratio is maximized when the column vectors of the projection matrix, *W*, are the eigenvectors of $S_W^{-1} S_B$. It should be noted that *W* maps the original d_1-dimensional data space $\boldsymbol{X}$ to a d_2-dimensional space Δ ($d_2 \le C-1$, *C* is the number of classes).

For both FDA and MDA, the columns of the optimal *W* are the generalized eigenvector(s) $\mathbf{w}_i$ associated with the largest eigenvalue(s) . $W_{opt} = [\mathbf{w}_1, \mathbf{w}_{2,} \cdots, \mathbf{w}_{C-1}]$ will contain in its columns *C-1* eigenvectors corresponding to *C-1* eigenvalues, i.e., $S_B w_i = \lambda_i S_W w_i$ [12].

2.2. Kernel Discriminant Analysis

To take into account non-linearity in the data, we propose a kernel-based approach. The original MDA algorithm is applied in a feature space *F* which is related to the original space by a non-linear mapping ϕ: $x \rightarrow \phi(x)$. Since in general the number of components in $\phi(x)$ can be very large or even infinite, this mapping is too expensive and will not be carried out explicitly, but through the evaluation of a kernel *k*, with elements $k(\mathbf{x}_{\mathrm{i}}, \mathbf{x}_{\mathrm{j}}) = \phi(\mathbf{x}_{\mathrm{i}})^T \cdot \phi(\mathbf{x}_{\mathrm{j}})$. This is the same idea adopted by the support vector machine [14], kernel PCA [29], and invariant feature extractions [30, 31]. The trick is to rewrite the MDA formulae using only dot products of the form $\phi_i^T \cdot \phi_j$, so that the reproducing kernel matrix can be substituted into the formulation and the solution; eliminate the need for direct nonlinear transformation.

Using superscript $^{\phi}$ to denote quantities in the new space and using S_B and S_W for between-class scatter matrix and within-class scatter matrix, we have the objective function in the following form:

$$W_{opt} = \arg\max_{W} \frac{|W^T S_B^\phi W|}{|W^T S_W^\phi W|} \tag{5}$$

and

$$S_B^\phi = \sum_{j=1}^{C} N_j \cdot (m_j^\phi - m^\phi)(m_j^\phi - m^\phi)^T \tag{6}$$

$$S_W^\phi = \sum_{j=1}^{C}\sum_{i=1}^{N_j} (\phi(\mathbf{x}_i^{(j)}) - m_j^\phi)(\phi(\mathbf{x}_i^{(j)}) - m_j^\phi)^T \tag{7}$$

with $m^\phi = \frac{1}{N}\sum_{k=1}^{N}\phi(\mathbf{x}_{\mathrm{k}})$, $m_j^\phi = \frac{1}{N_j}\sum_{k=1}^{N_j}\phi(\mathbf{x}_{\mathrm{k}})$ where $j = 1, \cdots, C$, and N is the total number of samples.

In general, there is no other way to express the solution $W_{opt} \in F$, either because F is too high or infinite dimension, or because we do not even know the actual *feature space* connected to a certain kernel. But we know [22, 24] that any column of the solution W_{opt} must lie in the span of all training samples in F, i.e., $\mathbf{w}_i \in F$. Thus for some expansion coefficients $\vec{\alpha} = [\alpha_1, \cdots, \alpha_N]^T$,

$$\mathbf{w}_i = \sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}_k) = \Phi\vec{\alpha} \qquad i = 1, \ldots, N \tag{8}$$

where $\Phi = [\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_N)]$. We can therefore project a data point $\mathbf{x}_k$ onto one coordinate of the linear subspace of F as follows (we will drop the subscript on $\mathbf{w}_i$ in the ensuing):

$$\mathbf{w}^T \phi(\mathbf{x}_k) = \vec{\alpha}^T \Phi^T \phi(\mathbf{x}_k) \tag{9}$$

$$= \vec{\alpha}^T \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_N, \mathbf{x}_k) \end{bmatrix} = \vec{\alpha}^T \xi_k \tag{10}$$

$$\xi_k = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_N, \mathbf{x}_k) \end{bmatrix} \tag{11}$$

where we have rewritten dot products, $\phi(\mathbf{x})^T \cdot \phi(\mathbf{y})$ with kernel notation $k(\mathbf{x},\mathbf{y})$. Similarly, we can project each of the class means onto an axis of the subspace of feature space F using only products:

$$\mathbf{w}^T m_j^\phi = \vec{\alpha}^T \tfrac{1}{N_j} \sum_{k=1}^{N_j} \begin{bmatrix} \phi(\mathbf{x}_1)^T \cdot \phi(\mathbf{x}_k) \\ \vdots \\ \phi(\mathbf{x}_N)^T \cdot \phi(\mathbf{x}_k) \end{bmatrix} \tag{12}$$

$$= \vec{\alpha}^T \begin{bmatrix} \tfrac{1}{N_j} \sum_{k=1}^{N_j} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ \tfrac{1}{N_j} \sum_{k=1}^{N_j} k(\mathbf{x}_N, \mathbf{x}_k) \end{bmatrix} \tag{13}$$

$$= \vec{\alpha}^T \boldsymbol{\mu}_j \tag{14}$$

It follows that

$$\mathbf{w}^T S_B \mathbf{w} = \vec{\alpha}^T K_B \vec{\alpha} \tag{15}$$

where $K_B = \sum_{j=1}^{C} N_j (\boldsymbol{\mu}_\mathbf{j} - \boldsymbol{\mu})(\boldsymbol{\mu}_\mathbf{j} - \boldsymbol{\mu})^T$ and

$$\mathbf{w}^T S_W \mathbf{w} = \vec{\alpha}^T K_W \vec{\alpha} \tag{16}$$

where $K_W = \sum_{j=1}^{C} \sum_{k=1}^{N_j} (\xi_k - \boldsymbol{\mu}_j)(\xi_k - \boldsymbol{\mu}_j)^T$. The goal of kernel multiple discriminant analysis (KMDA) is to find

$$\mathbf{A}_{opt} = \arg\max_{\mathbf{A}} \frac{|\mathbf{A}^T K_B \mathbf{A}|}{|\mathbf{A}^T K_W \mathbf{A}|} \tag{17}$$

where $\mathbf{A} = [\vec{\alpha}_1, \cdots, \vec{\alpha}_{C-1}]$, C is the total number of classes, N is the size of training samples, and K_B and K_W are $N \times N$ matrices which require only kernel computations on the training samples [22].

Now we can solve for $\vec{\alpha}$ sí, the projection of a new pattern z onto ***w*** is given by equations (9)-(10). Similarly, algorithms using different matrices for S_1, and S_2 in equation (1), are easily obtained along the same lines.

2.3. Biased Discriminant Analysis

BDA [13] differs from traditional MDA defined in equations (1)-(3) and (5)-(7) in a modification on the computation of between-class scatter matrix S_B and within-class scatter matrix S_W. They are replaced by $S_{N\to P}$ and S_P, respectively.

$$S_{N\to P} = \sum_{i=1}^{N_y} (\mathbf{y}_i - m_x)(\mathbf{y}_i - m_x)^T \qquad (18)$$

$$S_P = \sum_{i=1}^{N_x} (\mathbf{x}_i - m_x)(\mathbf{x}_i - m_x)^T \qquad (19)$$

where $\{\mathbf{x}_i, i=1,\cdots,N_x\}$ denotes the positive examples and $\{\mathbf{y}_i, i=1,\cdots,N_y\}$ denotes the negative examples, m_x is the mean vector of the sets $\{\mathbf{x}_i\}$, respectively. $S_{N\to P}$ is the scatter matrix between the negative examples and the centroid of the positive examples, and S_P is the scatter matrix within the positive examples. $N \to P$ indicates the asymmetric property of this approach, i.e., the userís biased opinion towards the positive class, thus the name of biased discriminant analysis [13].

2.4. Regularization and Discounting Factors

It is well known that sample-based plug-in estimates of the scatter matrices based on equations (2, 3, 6, 7, 18, 19) will be severely biased for small number of training samples, i.e., the largest eigenvalue becomes larger, while the small ones become smaller. If the number of the feature dimensions is large compared with the number of training examples, the problem becomes ill-posed. Especially in the case of kernel algorithms, we effectively work in the space spanned by all N mapped training examples $\phi(\mathbf{x})$ which are, in practice, often linearly dependent. For instance, for KMDA, a solution with zero within class scatter (i.e., $\mathbf{A}^T K_W \mathbf{A}$ =0) is very likely due to overfitting. A compensation or regulation can be done by adding small quantities to the diagonal of the scatter matrices [32].

3. TRAINING ON A SUBSET

We still have one problem: while we could avoid working explicitly in the extremely high or infinite dimensional space *F*, we are now facing a problem in *N* variables, a number which in many practical applications would not allow to store or manipulate $N \times N$ matrices on a computer anymore. Furthermore, solving an eigen-problem or a QP of this size is very time consuming ($O(N^3)$). To maximize equation (17), we need to solve an $N \times N$ eigen- or mathematical programming problem, which might be intractable for large *N*. Approximate solutions could be obtained by sampling representative subsets of the training data $\{\mathbf{x}_k \mid k = 1, \cdots, M, \ \ M << N\}$, and using $\grave{\boldsymbol{\xi}}_k = [k(\mathbf{x}_1, \mathbf{x}_k), \cdots, k(\mathbf{x}_M, \mathbf{x}_k)]^T$ to take the place of $\boldsymbol{\xi}_k$. Two data-sampling schemes are proposed.

3.1. PCA-based Kernel Vector Selection

The first scheme is blind to the class labeling. We select representatives, or *kernel vectors*, by identifying those training samples which are likely to play a key role in $\Xi = [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_N]$. Ξ is a $N \times N$ matrix, but $rank(\Xi) << N$, when the size of training dataset is very large. This fact suggests that some training samples could be ignored in calculating kernel features $\boldsymbol{\xi}$.

We first compute the principal components of Ξ. Denote the $N \times N$ matrix of concatenated eigenvectors with **P**. Thresholding elements of **abs**(**P**) by some fraction of the largest element of it allows us to identify salient PCA coefficients. For each column corresponding to a non-zero eigenvalue, choose the training samples which correspond to a salient PCA coefficient, i.e., choose the training samples corresponding to rows that survive the thresholding. Do so for every non-zero eigenvalue and we arrive at a decimated training set, which represents data at the periphery of each data cluster.

Figure 1 shows an example of KMDA with 2D two-class non-linear-separable samples.

3.2 Evolutionary Kernel Vector Selection

The second scheme is to take advantage of class labels in the data. We maintain a set of kernel vectors at every iteration which are meant to be the key pieces of data for training. *M* initial kernel vectors, $KV^{(0)}$, are chosen at random. At iteration *k*, we have a set of kernel vectors, $KV^{(k)}$, which are used to perform KMDA such that the nonlinear projection $\mathbf{y}_i^{(k)} = \mathbf{w}^{(k)^T} \phi(\mathbf{x}_i) = \mathbf{A}_{opt}^{(k)^T} \boldsymbol{\xi}_i^{(k)} \in \Delta$ of the original data $\mathbf{x}_i$ can be obtained. We assume Gaussian distribution $\theta^{(k)}$ for each class in the nonlinear discrimination space Δ, and the parameters $\theta^{(k)}$ can be estimated by $\{\mathbf{y}^{(k)}\}$, such that the labeling and training error $e^{(k)}$ can be obtained by $\bar{l}_i^{(k)} = \arg\max_j p(l_j | \mathbf{y}_i, \theta(k)\}$.

If $e^{(k)} < e^{(k-1)}$, we randomly select *M* training samples from the correctly classified training samples as kernel vector $KV^{(t+1)}$ at iteration $k+1$. Another possibility is that if any current kernel vector is correctly classified, we randomly select a sample in its topological neighborhood to replace this kernel vector in the next iteration. Otherwise, i.e., $e^{(k)} > e^{(k-1)}$, and we terminate.

The evolutionary kernel vector selection algorithm is summarized below:

Evolutionary Kernel Vector Selection: Given a set of training data $D = (\mathbf{X}, L) = \{(\mathbf{x}_i, l_i),\ i = 1, \cdots, N\}$ to identify a set of *M* kernel vectors $KV = \{v_i, i = 1, \cdots, M\}$.

```
$k = 0;\ e = \infty;\ KV^{(0)} = random\_pick(\mathbf{X});$ // Init
do {
        $\mathbf{A}_{opt}^{(k)} = KMDA(\mathbf{X}, KV^{(k)})$;   // Perform KMDA
        $Y^{(k)} = Proj(\mathbf{X}, \mathbf{A}_{opt}^{(k)})$;          // Project X to $\Delta$
        $\theta^{(k)} = Bayes(Y^{(k)}, L)$;          // Bayesian Classifier
        $\overline{L}^{(k)} = Labeling(Y^{(k)}, \theta^{(k)})$;  // Classification
        $e^{(k)} = Error(\overline{L}^{(k)}, L)$;          // Calculate error
        if ( $e^{(k)} < e$ )
                $e = e^{(k)};\ KV = KV^{(k)};\ k++;$
                $KV^{(k)} = random\_pick(\{x_i : \bar{l}_i^{(k)} \neq l_i\})$;
        else
                $KV = KV^{(k-1)}$;
                break;
        end
}
return $KV$;
```

4. KERNEL DEM ALGORITHM

In this paper, image classification is formulated as a *transductive* problem, which is to generalize the mapping function learned from the labeled training data set *L* to a specific unlabeled data set *U*. We make an assumption here that *L* and *U* are from the same distribution. This assumption is reasonable because in content-based image retrieval, the query images are drawn from the same image database. In short, image retrieval is to classify the images in the database by:

$$y_i = \arg \max_{j=1,\cdots,C} p(y_j \mid \mathbf{x}_i, L, U : \forall \mathbf{x}_i \in U\} \qquad (20)$$

where *C* is the number of classes and y_i is the class label for $\mathbf{x}_i$.

The expectation-maximization (EM) [12] approach can be applied to this transductive learning problem, since the labels of unlabeled data can be treated as missing values. We assume that the hybrid data set is drawn from a mixed density distribution of *C* components $\{c_j, j = 1, \cdots, C\}$, which are parameterized by $\Theta = \{\theta_j, j = 1, \cdots, C\}$. The mixture model can be represented as:

$$p(\mathbf{x} \mid \Theta) = \sum_{j=1}^{C} p(\mathbf{x} \mid c_j; \theta_j) p(c_j \mid \theta_j) \qquad (21)$$

where $\mathbf{x}$ is sample drawn from the hybrid data set $D = L \cup U$. We make another assumption that each component in the mixture model corresponds to one class, i.e., $\{y_j = c_j, j = 1, \cdots, C\}$.

Since the training data set D is union of labeled data set L and unlabeled data set U, the joint probability density of the hybrid data set can be written as:

$$p(D \mid \Theta) = \prod_{\mathbf{x}_i \in U} \sum_{j=1}^{C} p(c_j \mid \Theta) p(\mathbf{x}_i \mid c_j ; \Theta) \cdot \prod_{\mathbf{x}_i \in L} p(y_i = c_i \mid \Theta) p(\mathbf{x}_i \mid y_i = c_i ; \Theta) \qquad (22)$$

The above equation holds when we assume that each sample is independent to others. The first part of equation (22) is for the unlabeled data set, and the second part is for the labeled data set.

The parameters Θ can be estimated by maximizing *a posteriori* probability $p(\Theta \mid D)$. Equivalently, this can be done by maximizing $\log(p(\Theta \mid D))$. Let $l(\Theta \mid D) = \log(p(\Theta) p(D \mid \Theta))$, and we have

$$l(\Theta \mid D) = \log(p(\Theta)) + \sum_{\mathbf{x}_i \in U} \log(\sum_{j=1}^{C} p(c_j \mid \Theta) p(\mathbf{x}_i \mid c_j ; \Theta)) + \sum_{\mathbf{x}_i \in L} \log(p(y_i = c_i \mid \Theta) p(\mathbf{x}_i \mid y_i = c_i ; \Theta)) \quad (23)$$

Since the log of sum is hard to deal with, a binary indicator $\mathbf{z}_i$ is introduced, $\mathbf{z}_i = (z_{i1}, \cdots, z_{iC})$, denoted with observation O_j: $z_{ij} = 1$ if and only if $y_i = c_j$, and $z_{ij} = 0$ otherwise, so that

$$l(\Theta \mid D, Z) = \log(p(\Theta)) + \sum_{\mathbf{x}_i \cup D} \sum_{j=1}^{C} z_{ij} \log(p(O_j \mid \Theta) p(\mathbf{x}_i \mid O_j ; \Theta)) \qquad (24)$$

The EM algorithm can be used to estimate the probability parameters Θ by an iterative hill climbing procedure, which alternatively calculates $E(Z)$, the expected values of all unlabeled data, and estimates the parameters Θ given $E(Z)$. The EM algorithm generally reaches a local maximum of $l(\Theta \mid D)$.

As an extension to EM algorithm, [11] proposed a three step algorithm, called Discriminant-EM (DEM), which loops between an expectation step, a discriminant step (via MDA), and a maximization step. DEM estimates the parameters of a generative model in a discriminating space.

As discussed in Section 2.2, Kernel DEM (KDEM) is a generalization of DEM in which instead of a simple linear transformation to project the data into discriminant subspaces, the data is first projected nonlinearly into a high dimensional feature space F where the data is better linearly separated. The nonlinear mapping $\phi(\cdot)$, is implicitly determined by the kernel function, which must be determined in advance. The transformation from the original data space $\mathbf{X}$ to the discriminating space Δ, which is a linear subspace of the feature space *F*, is given by $\mathbf{w}^T\phi(\cdot)$ implicitly or $\mathbf{A}^T\xi$ explicitly. A low-dimensional generative model is used to capture the transformed data in Δ.

$$p(y\,|\,\Theta)=\sum_{j=1}^{C}p(\mathbf{w}^T\phi(\mathbf{x})\,|\,c_j;\theta_j)p(c_j\,|\,\theta_j) \qquad (25)$$

Empirical observations suggest that the transformed data *y* approximates a Gaussian in Δ, and so in our current implementation, we use low-order Gaussian mixtures to model the transformed data in Δ. Kernel DEM can be initialized by selecting all labeled data as kernel vectors, and training a weak classifier based on only labeled samples. Then, the three steps of Kernel DEM are iterated until some appropriate convergence criterion:

- E-step: set $\hat{Z}^{(k+1)}=E[Z\,|\,D;\hat{\Theta}^{(k)}]$
- D-step: set $\mathbf{A}_{opt}^{k+1}=\arg\max_{A}\dfrac{|\mathbf{A}^TK_B\mathbf{A}|}{|\mathbf{A}^TK_W\mathbf{A}|}$, and project a data point $\mathbf{x}$ to a linear subspace of feature space *F*.
- M-Step: set $\hat{\Theta}^{(k+1)}=\arg\max_{\Theta}p(\Theta\,|\,D;\hat{Z}^{(k+1)})$

The E-step gives probabilistic labels to unlabeled data, which are then used by the D-step to separate the data. As mentioned above, this assumes that the class distribution is moderately smooth.

5. EXPERIMENTS AND ANALYSIS

In this section, we compare KMDA and KDEM with other supervised learning techniques on various benchmark datasets and synthetic data for several image classification tasks.

5.1. Benchmark Test for KMDA

In the first experiment, we verify the ability of KMDA with our data sampling algorithms. Several benchmark datasets[2] are used in the experiments. For comparison, KMDA is compared with a single RBF classifier (RBF), a support vector machine (SVM), AdaBoost, and the kernel Fisher discriminant (KFD) on the benchmark dataset [33] and linear MDA. Kernel functions that have been proven useful are e.g., Gaussian RBF, $k(\mathbf{x},\mathbf{z}) = \exp(- \|\mathbf{x} - \mathbf{z}\|^2 / c)$, or polynomial kernels, $k(\mathbf{x},\mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^{\mathrm{d}}$, for some positive constants $c \in R$ and $d \in N$, respectively [22]. In this work, Gaussian RBF kernels are used in all kernel-based algorithms.

In Table 1, KMDAñrandom is KMDA with kernel vectors randomly selected from training samples, KMDAñpca is KMDA with kernel vectors selected from training samples based on PCA, KMDAñ evolutionary is KMDA with kernel vectors selected from training samples based on evolutionary scheme. The benchmark test shows the Kernel MDA achieves comparable performance as other state-of-the-art techniques over different training datasets, in spite of the use of a decimated training set. Comparing three schemes of selecting kernel vectors, it is clear that both PCA-based and evolutionary-based schemes work slightly better than the random selection scheme by having smaller error rate and/or smaller standard deviation. Finally, Table 1 clearly shows that superior performance of KMDA over linear MDA.

5.2. Kernel Setting

There are two parameters need to be determined for kernel algorithms using Gaussian RBF kernel. The first is the variance c and the second is the number of kernel vectors used. Till now there is no good method on how to choose a kernel function and its parameter. In the second experiment, we will determine variance c and number of kernel vectors empirically using Gaussian RBF kernel as an example. The same benchmark dataset as in Section 5.1 is used. Figure 2 shows the average error rate in percentage of KDEM with Gaussian RBF kernel under different variance c and varying number of kernel vectors used on Heart data. By empirical observation, we find that 10 for *c* and 20 for number of kernel vectors gives nearly the best performance at a relatively low computation cost. Similar results are obtained for tests on Breast-Cancer data and Banana data. Therefore this kernel setting will be used in the rest of our experiments.

5.3. Image Classification on Corel Photos

In the third experiment, a series of classification methods are compared in photo classification scenario. Those methods are MDA, BDA, DEM, KMDA, KBDA, and KDEM. The stock photos are selected from widely used COREL dataset. Corel photos covers a wide range of more than 500 categories, ranging from animals and birds to Tibet and the Czech Republic. In this test, we use Corel photos consisting of 15 classes with 99 images in each class. Randomly selected 4 classes of images are used in the experiment. One class of images is considered as positive while all the other 3 classes are considered as negative. The training set consists of 40 to 80 images with half positive and half negative images. The rest images in the database (1485) are all used as testing set.

We will first describe the visual features used. There are three visual features used: color moments [34], wavelet-based texture [35], and water-filling edge-based structure feature [36]. The color space we use is HSV because of its de-correlated coordinates and its perceptual uniformity. We extract the first three moments (mean, standard deviation and skewness) from the three-color channels and therefore have a color feature vector of length $3\times 3=9$. For wavelet-based texture, the original image is fed into a wavelet filter bank and is decomposed into 10 de-correlated sub-bands. Each sub-band captures the characteristics of a certain scale and orientation of the original image. For each sub-band, we extract the standard deviation of the wavelet coefficients and therefore have a texture feature vector of length 10. For water-filling edge-based structure feature vector, we first pass the original images through an edge detector to generate their corresponding edge map. We extract eighteen (18) elements from the edge maps, including *max fill time, max fork count, etc.* For a complete description of this edge feature vector, interested readers

[2] The benchmark data sets are obtained from http://mlg.anu.edu.au/~raetsch/

are referred to [36]. In our experiments, the 37 visual features (9 color moments, 10 wavelet moments and 18 water-filling features) are pre-extracted from the image database and stored off-line.

Table 2 shows the comparison of different classification methods using MDA, BDA, DEM and their kernel algorithms. There are several conclusions: (1) The classification error drops with the increasing size of the training dataset for all methods; (2) All the kernel algorithms perform better than their linear algorithms. This shows that the kernel machine has better capacity than linear approaches to separate linearly non-separable data. (3) KDEM performs best (comparable to KMDA for the training size 80) among all the methods. This shows that the incorporation of unlabeled data and nonlinear mapping contributes to the improved performance. (4) DEM performs better than MDA. This shows that incorporation of unlabeled data in semi-supervised learning in DEM helps improve classification accuracy. (5) The results for BDA and DEM are mixed but KDEM performs better than KBDA. We will further compare KBDA and KDEM in the next section.

As to the computational complexity of each algorithm, Table 2 also shows the actually executing time to run each algorithm. The Matlab program was running on a Pentium IV processor of 2.4 GHz CPU with 256 MB memory. KDEM is most computationally complex among all the algorithms. But the extra computational complexity introduced by semi-supervised learning, i.e, EM, and the nonlinear mapping is still affordable, e.g., within 3 seconds, for the training data size 80 and testing data size 1485.

Finally a summary of the advantages and disadvantages of the discriminant algorithms for MDA, BDA, DEM, KMDA, KBDA, and KDEM is given in Table 3.

5.4. KDEM versus KBDA for Image Classification

As reported in [13], biased discriminant analysis (BDA) has achieved satisfactory results in content-based image retrieval when the number of training samples is small (<20). BDA differs from traditional MDA in that it tends to cluster all the positive samples and scatter all the negative samples from the centroid of the positive examples. This works very well with relatively small training set. However, BDA is biased tuned towards the centroid of the positive examples. It will be effective only if these positive examples are the *most-informative* images [3, 4], e.g., images close to the classification boundary. If the positive examples are *most-positive* images [3, 4], e.g., the images far away from the classification boundary. The optimal transformation found based on the *most-positive* images will not help the classification for images on the boundary. Moreover, BDA ignores the unlabeled data and takes only the labeled data in the learning.

In the fourth experiment, Kernel DEM (KDEM) is further compared with the Kernel BDA (KBDA) on both image database and synthetic data. Figure 3 shows the average classification error rate in percentage for KDEM and KBDA with the same RBF kernel for face and non-face classification. The face images are from MIT facial image database[3] [37] and non-face images are from Corel database. There are 2429 faces images from MIT database. There are 1485 non-face images (15 classes with about 99 in each class) from Corel database. Some examples of face and non-face images are shown in Figure 4. For training dataset, 100 face images are randomly selected from MIT database, and a varying number of 5 to 200 non-face images are randomly selected from Corel database. The testing dataset consists of 200 randomly images (100 faces and 100 non-faces) from two databases. The images are resized to 16×16 and converted to a column-wise concatenated feature vector.

In Figure 3, when the size of negative examples is small (<20), KBDA outperforms KDEM while KDEM performs better when more negative examples are provided. This agrees with our expectation.

In the above experiment, the size of negative examples is increased from 5 to 200. There is a possibility that most of the negative examples are from the same class. To further test the capability of KDEM and KBDA in classifying negative examples with varying number of classes, we perform experiments on synthetic data for which we have more controls over data distribution.

A series of synthetic data is generated based on Gaussian or Gaussian mixture models with feature dimension of 2, 5 and 10 and varying number of negative classes from 1 to 9. In the feature space, the

[3] MIT facial database can be downloaded from http://www.ai.mit.edu/projects/cbcl/software-datasets/FaceData2.html

centroid of positive samples is set at origin and the centroids of negative classes are set randomly with distance 1 to the origin. The variance of each class is a random number between 0.1 and 0.3. The features are independent to each other. We include 2D synthetic data for visualization purpose. Both the training and testing sets have fixed size of 200 samples with 100 positive samples and 100 negative samples with varying number of classes.

Figure 5 shows the comparison of KDEM, KBDA and DEM algorithms on 2-D, 5-D and 10-D synthetic data. In all cases, with the increasing size of negative classes from 1 to 9, KDEM always performs better than KBDA and DEM thus shows its superior capability of multi-class classification. Linear DEM has comparable performance with KBDA on 2-D synthetic data and outperforms KBDA on 10-D synthetic data. One possible reason is that learning is on hybrid data in both DEM and KDEM, while only labeled data is used in KBDA. This indicates that proper incorporation of unlabeled data in semi-supervised learning does improve classification to some extent.

5.5. Correlated versus Independent Features

Feature independence is usually assumed in high-dimensional feature space for training samples. However, this assumption is not always true in practice. Such an example is color correlogram [38], a feature commonly used in content-based image retrieval.

In the fifth experiment, we will investigate how feature correlation will affect algorithm performance in two-class classification problem such as image retrieval. Two classes of samples, i.e., positive and negative are synthesized based on the Gaussian mixture model (*C=2*) with feature dimension 10, variance 1 and distance between the centroids of two classes 2. How to generating correlated features is referred to [39]

Table 4 shows the average error rate with different pair-wise correlation coefficients ρ among positive examples and negative examples. For example, the pair (0.1, 0.6) means the correlation among all pair-wise feature components is 0.1 for the negative examples and 0.6 for the positive examples. From table 2, the corresponding error rate of KDEM is 4.05%.

It is clear when features are highly correlated, it becomes easier for Kernel DEM algorithm to separate the positive examples from the negative examples. Because data is more clustered with larger correlation. Similar results are obtained for linear DEM algorithm. This is reasonable and an illustration example can be visualized in Figure 6. c_1 and c_2 represent the centroids of 2D two-class samples, respectively. When the features of two classes are independent to each other, their distributions can be illustrated by two circles in blue and red, respectively. Using discriminant analysis, the optimal transformation will be projection 1 onto subspace 1 (i.e., 1D space). However, two classes will still have some overlap between *B* and *C* along the projected 1D subspace and this will result in classification error. But when the features of two classes are highly correlated, the distributions of two classes are illustrated by two ellipses in blue and red, respectively. The optimal transformation found by discriminant analysis will be projection 2. In this case, the projected samples of two classes have no overlap and can be easily separated in 1D subspace.

5.6. Hand Posture Recognition

In the sixth experiment, we examine KDEM on a hand gesture recognition task. The task is to classify among 14 different hand postures, each of which represents a gesture command model, such as navigating, pointing, grasping, etc. Our raw dataset consists of 14,000 unlabeled hand images together with 560 labeled images (approximately 40 labeled images per hand posture), most from video of subjects making each of the hand postures. These 560 labeled images are used to test the classifiers by calculating the classification errors.

Hands are localized in video sequences by adaptive color segmentation and hand regions are cropped and converted to gray-level images. Gabor wavelet [40] filters with 3 levels and 4 orientations are used to extract 12 texture features. 10 coefficients from the Fourier descriptor of the occluding contour are used to represent hand shape. We also use area, contour length, total edge length, density, and 2nd moments of edge distribution, for a total of 28 low-level image features (I-feature). For comparison, we represent images by coefficients of 22 largest principal components of the dataset resized to 20 × 20 pixels (these are ìeigenimagesî, or E-features). In out experiments, we use 140, i.e., 10 for each hand posture, and 10,000 (randomly selected from the whole database) labeled and unlabeled images respectively, for training both EM and DEM.

Table 5 shows the comparison. Six classification algorithms are compared in this experiment. The multi-layer perceptron [41] used in this experiment has one hidden layer of 25 nodes. We experiment with two schemes of the nearest neighbor classifier. One uses just 140 labeled samples, and the other uses these 140 labeled samples to bootstrap the classifier by a growing scheme, in which newly labeled samples will be added to the classifier according to their labels.

We observe that multi-layer perceptrons are often trapped in local minima and nearest neighbors suffers from the sparsity of the labeled templates. The poor performance of pure EM is due to the fact that the generative model does not capture the ground-truth distribution well, since the underlying data distribution is highly complex. It is not surprising that linear DEM and KDEM outperform other methods, since the D-step optimizes separability of the class.

Comparing KDEM with DEM, we find KDEM often appears to project classes to approximately Gaussian clusters in the transformed spaces, which facilitate their modeling with Gaussians. Figure 7 shows typical transformed data sets for linear and nonlinear discriminant analysis, in a projected 2D subspaces of three different hand postures. Different postures are more separated and clustered in the nonlinear subspace by KMDA. Figure 8 shows some examples of correctly classified and mislabeled hand postures for KDEM and linear DEM.

6. CONCLUSIONS

Two sampling schemes are proposed for efficient, kernel-based, nonlinear, multiple discriminant analysis. These algorithms identify a representative subset of the training samples for the purpose of classification.

Benchmark tests show that KMDA with these adaptations not only outperforms the linear MDA but also performs comparably with the best known supervised learning algorithms. We also present a self-supervised discriminant analysis technique, Kernel DEM (KDEM), which employs both labeled and unlabeled data in training. On real image database and synthetic data for several applications of image classification, KDEM shows the superior performance over biased discriminant analysis (BDA), naÔve supervised learning and existing semi-supervised learning algorithms.

Our future work includes several aspects. (1) We will look into advanced regularization schemes for the discriminant-based approaches; (2) We will intelligently integrate biased discriminant analysis with multiple discriminant analysis in a unified framework so that their advantages can be utilized and disadvantages can be compensated by each other. (3) To avoid the heavy computation over the whole database, we will investigate schemes of selecting a representative subset of unlabeled data whenever unlabeled data helps. (4) Gaussian or Gaussian mixture models are assumed for data distribution in the projected optimal subspace, even when the initial data distribution is highly non-Gaussian. We will examine the data modeling issue more closely with Gaussian (or Gaussian mixture) and non-Gaussian distributions.

ACKNOWLEDGEMENT

This work was supported in part by University of Texas at San Antonio, and by National Science Foundation (NSF) under the grant IIS-03-08222 in Northwestern University.

References

1. A. Smeulder, M. Worring, S. Santini, A. Gupta, and R. Jain, ìContent-based image retrieval at the end of the early years,î *IEEE Trans. PAMI*, pp. 1349-1380, December 2000.
2. K. Tieu, and P. Viola, ìBoosting image retrieval,î *Proc. of IEEE Conf. CVPR*, June 2000.

3. I. J. Cox, M. L. Miller, T. P. Minka, and T. V. Papsthomas, ìThe Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments,î *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 20-37, 2000.

4. S. Tong, and E. Wang, ìSupport vector machine active learning for image retrieval,î *Proc. of ACM Intíl. Conf. Multimedia*, pp. 107-118, Ottawa. Canada, Oct. 2001.

5. Q. Tian, P. Hong, and T. S. Huang, ìUpdate relevant image weights for content-based image retrieval using support vector machinesî, *Proc. of IEEE Intíl Conf. Multimedia and Expo,* vol. 2, pp. 1199-1202, , NY, 2000.

6. Y. Rui, and T. S. Huang, ìOptimal learning in image retrieval,î *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 236-243, South Carolina, June 2000.

7. N. Vasconcelos, and A. Lippman, ìBayesian relevance feedback for content-based image retrieval,î *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, *CVPRí00*, Hilton Head Island, June 2000.

8. C. Yang, and T. Lozano-PÈrez, ìImage database retrieval with multiple-instance learning techniques,î *Proc. of the 16th Intíl Conf. Data Engineering*, Santa Diego, CA, 2003.

9. J. Laakaonen, M. Koskela, and E. Oja, ìPicSOM: self-organizing maps for content-based image retrieval,î *Proc. IEEE Intíl Conf. Neural Networks*, Washington, DC, 1999.

10. Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, ìRelevance feedback: a power tool in interactive content-based image retrievalî, *IEEE Trans Circuits and Systems for Video Tech.*, vol. 8, no. 5, pp. 644-655, Sept. 1998.

11. Y. Wu, Q. Tian, and T. S. Huang, ìDiscriminant EM algorithm with application to image retrieval,î *Proc. of IEEE Conf. Computer Vision and Pattern Recognition,* South Carolina, June 13-15, 2000.

12. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd edition, John Wiley & Sons, Inc., 2001.

13. X. Zhou, and T.S. Huang, ìSmall sample learning during multimedia retrieval using biasMap,î *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, Hawaii, December 2001.

14. V. Vapnik, *The nature of statistical learning theory*, second edition, Springer-Verlag, 2000.

15. F. G. Cozman, and I. Cohen, ìUnlabeled data can degrade classification performance of generative classifiers,î *Proc. of 15th Intíl Florida Artificial Intelli. Society Conf.*, pp. 327-331, Florida, 2002.

16. K. Nigram, A. K. McCallum, S. Thrun, and T. M. Mitchell, ìText classification from labeled and unlabeled documents using EM,î *Machine Learning*, 39(2/3):103-134, 2000.

17. T. Mitchell, ìThe role of unlabeled data in supervised learning,î *Proc. Sixth Intíl Colloquium on Cognitive Science*, Spain, 1999.

18. Cohen, N. Sebe, F. G. Cozman, M. C. Cirelo, and T. S. Huang, ìLearning Bayesian network classifiers for facial expression recognition with both labeled and unlabeled data,î *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, Madison, WI, 2003.

19. L. Wang, K. L. Chan, and Z. Zhang, ìBootstrapping SVM active learning by incorporating unlabelled images for image retrieval,î *Proc. of IEEE Conf. CVPR,* Madison, WI, 2003.

20. Y. Wu, and T. S. Huang, ìSelf-supervised learning for object recognition based on kernel discriminant-EM algorithm,î *Proc. of IEEE International Conf. on Computer Vision*, Canada, 2001.

21. Q. Tian, J. Yu, Y. Wu, and T.S. Huang, ìLearning based on kernel discriminant-EM algorithm for image classification,î *IEEE Intíl Conf. on Acoustics, Speech, and Signal Processing (ICASSP2004)*, May 17-24, 2004, Montreal, Quebec, Canada.

22. B. Schˆlkopf and A. J. Smola, *Learning with Kernels*. Mass: MIT Press, 2002.

23. L. Wolf, and A. Shashua, ìKernel principle for classification machines with applications to image sequence interpretation,î *Proc. of IEEE Conf. CVPR*, Madison, WI, 2003.

24. S. Mika, G. R‰tsch, J. Weston, B. Schˆlkopf, A. Smola, and K. M¸ller, ìConstructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces,î *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, May 2003.

25. R. A. Fisher, ìThe use of multiple measurement in taxonomic problems,î *Annals of Eugenics*, vol. 7, pp.179-188, 1936.

26. K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks*, New York: Wiley, 1996.

27. A. M. Martinez, and A. C. Kak, ìPCA versus LDA,î *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 23, No. 2, pp. 228-233, February 2001.

28. R. A. Fisher, ìThe statistical utilization of multiple measurements,î *Annals of Eugenics*, Vol. 8, pp. 376-386, 1938.

29. B. Schˆlkopf, A. Smola, and K. R. M¸ller, ìNonlinear component analysis as a kernel eigenvalue problem,î *Neural Computation*, vol. 10, pp. 1299-1319, 1998.

30. S. Mika, G. R‰tsch, J. Weston, B. Schˆlkopf, A. Smola, and K. R. M¸ller, ìInvariant feature extraction and classification in kernel spaces,î *Proc. of Neural Information Processing Systems*, Denver, Nov. 1999.

31. V. Roth, and V. Steinhage, ìNonlinear discriminant analysis using kernel functions,î *Proc. of Neural Information Processing Systems*, Denver, Nov. 1999.

32. J. Friedman, ìRegularized discriminant analysis,î *Journal of American Statistical Association*, vol. 84, no. 405, pp. 165-175, 1989.

33. S. Mika, G. R‰tsch, J. Weston, B. Schˆlkopf, A. Smola, and K. M¸ller, ìFisher discriminant analysis with Kernels,î *Proc. of IEEE Workshop on Neural Networks for Signal Processing*, 1999.

34. M. Stricker, and M. Orengo, ìSimilarity of color images,î *Proceedings of. SPIE Storage and Retrieval for Image and Video Databases*, San Diego, CA, 1995.

35. J. R. Smith, S. F. Chang, ìTransform features for texture classification and discrimination in large image database,î *Proceedings of IEEE Intíl Conference on Image Processing*, Austin, TX, 1994.

36. X. S. Zhou, Y. Rui, and T. S. Huang, ìWater-filling algorithm: a novel way for image feature extraction based on edge maps,î *Pros. of IEEE Intíl Conf. on Image Processing*, Kobe, Japan, 1999.

37. CBCL Face Database #1, MIT Center For Biological and Computation Learning
download from http://www.ai.mit.edu/projects/cbcl.

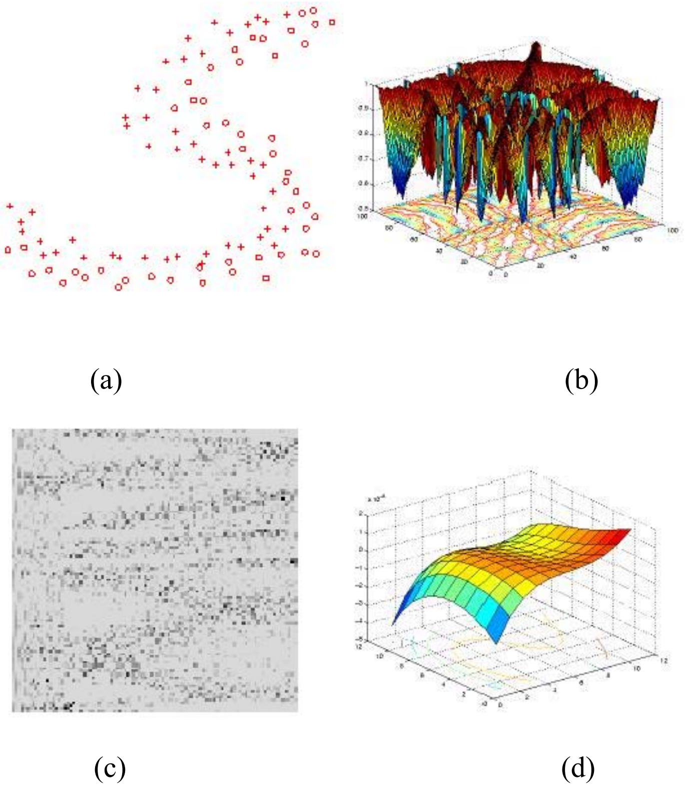
38. J. Huang, S. R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih, ìImage indexing using color correlogram,î *Proc. of IEEE Conf. CVPR*, pp. 762-768, June 1997.

39. M. Hugh, ìGenerating correlated random variables and stochastic process,î *Lecture Notes* 5, *Monte Carlo Simulation IEOR E4703*, Columbia University, March 2003.

40. A. K. Jain, and F. Farroknia, ìUnsupervised texture segmentation using Gabor filters,î *Pattern Recognition*, Vol. 24, No. 12, pp. 1167-1186, 1991.

41. S. Haykin, *Neural Networks*: *A Comprehensive Foundation*, second edition, Prentice Hall, New Jersey, 1999.

Note: The figures in this paper are better viewed in color under http://www.cs.utsa.edu/~qitian/pr_special/


(a) (b)

(c) (d)

Figure 1. KMDA with a 2D two-class nonlinear-separable example. (a) Original data (b) The kernel features of the data (c) The normalized coefficients of PCA on Ξ , in which only a small number of them are large (in black) (d) The nonlinear mapping

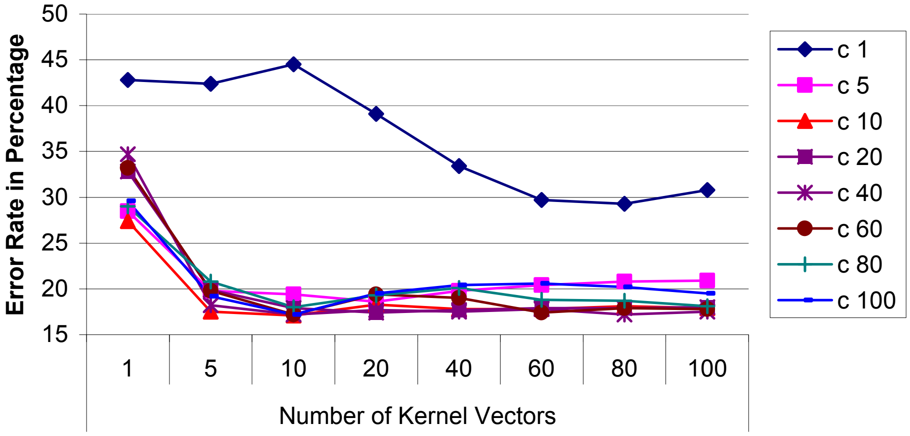


Figure 2. The average error rate for KDEM with Gaussian RBF kernel under varying variance *c* and number of kernel vectors on Heart data

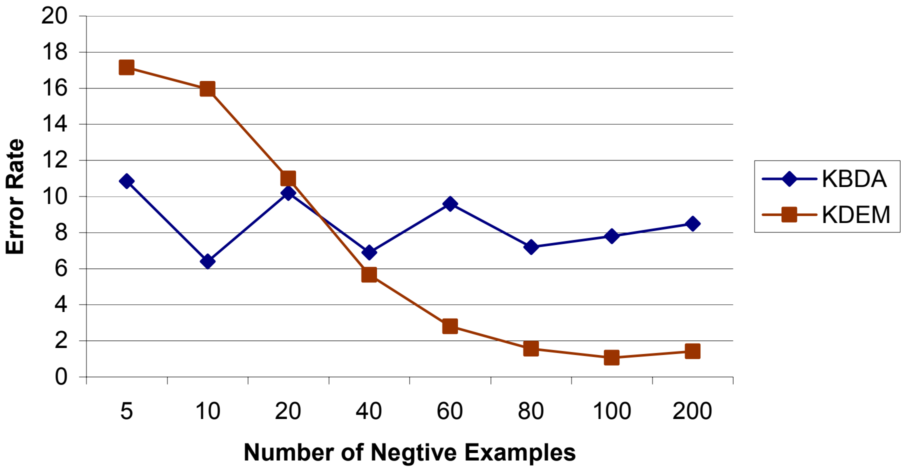
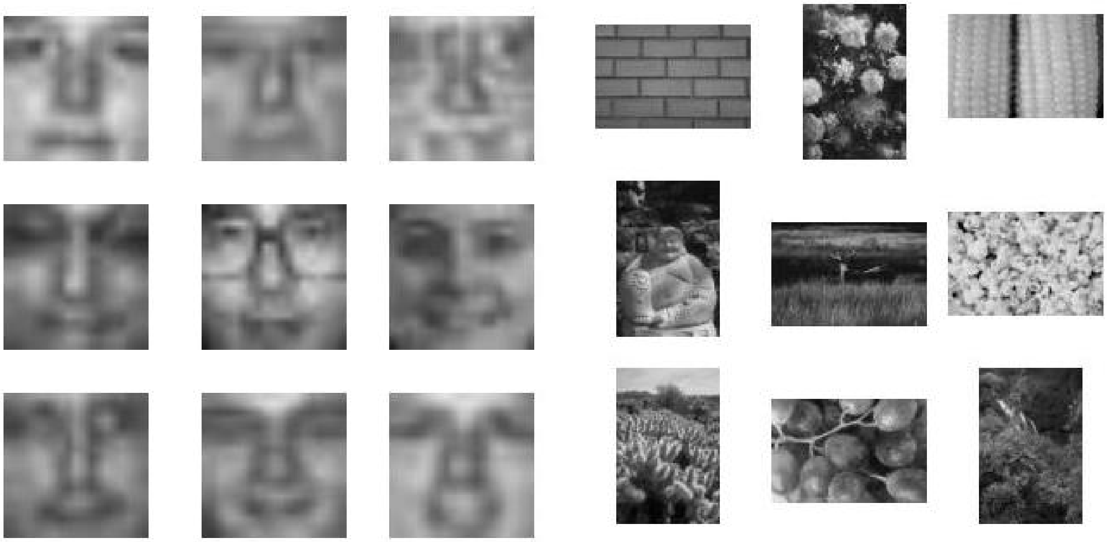


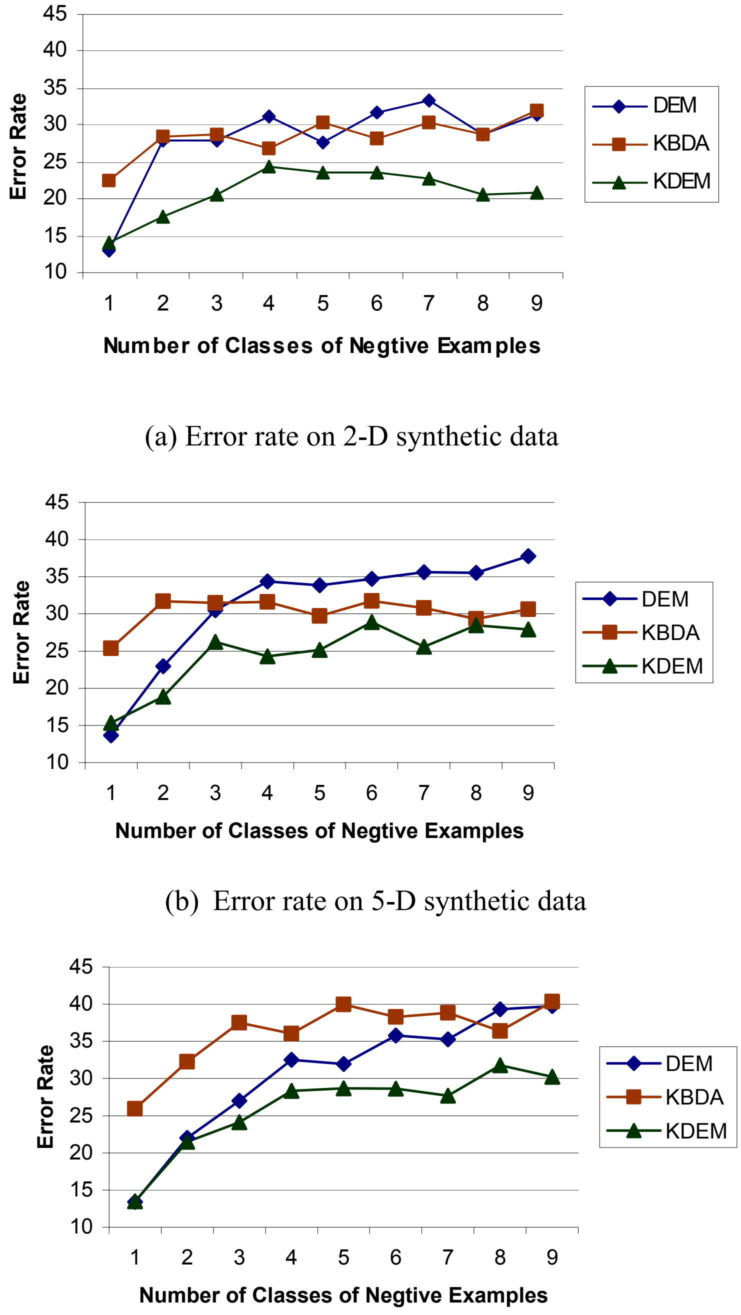
Figure 3. Comparison of KDEM and KBDA for face and non-face classification



(a) Face images from MIT facial database

(b) Non-face images from Corel database

Figure 4. Examples of (a) face images from MIT facial database and (b) non-face images from Corel database



(a) Error rate on 2-D synthetic data

(b) Error rate on 5-D synthetic data

(c) Error rate on 10-D synthetic data

Figure 5. Comparison of KDEM, KBDA and DEM algorithms on (a) 2-D (b) 5-D (c) 10-D synthetic data with varying number of negative classes

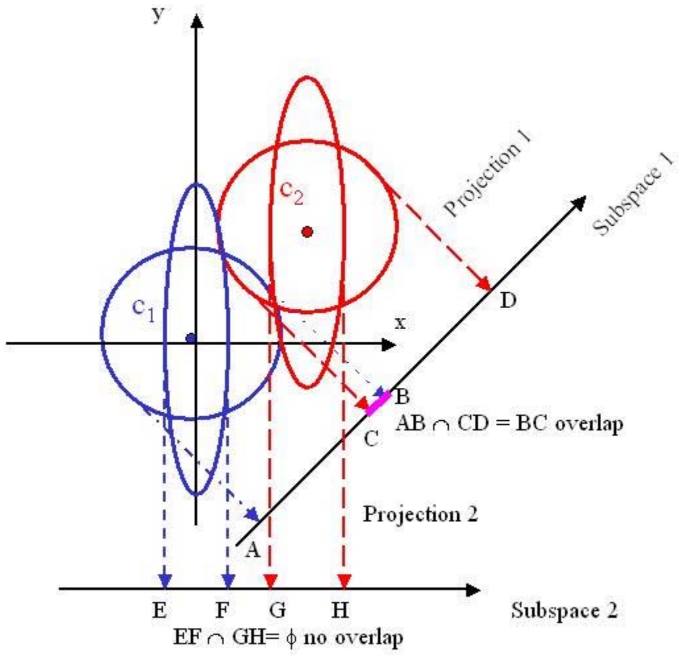
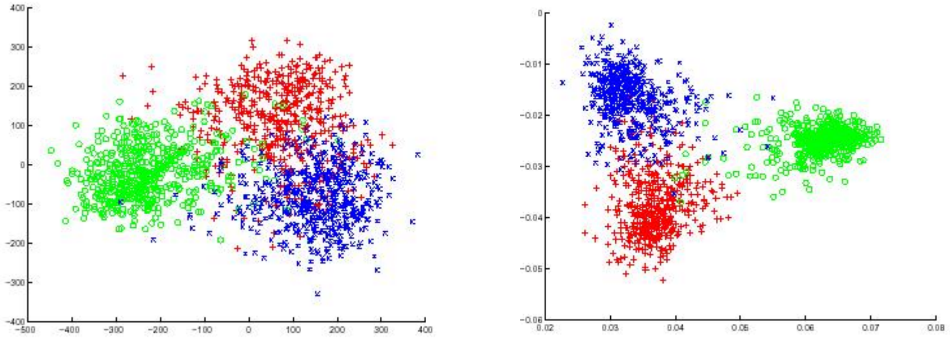


Figure 6. An illustration example of the optimal 1D subspace projection of correlated and independent features for 2D two-class samples



(a) (b)

Figure 7. Data distribution in the projected subspace (a) Linear MDA (b) Kernel MDA. Different postures are more separated and clustered in the nonlinear subspace by KMDA.

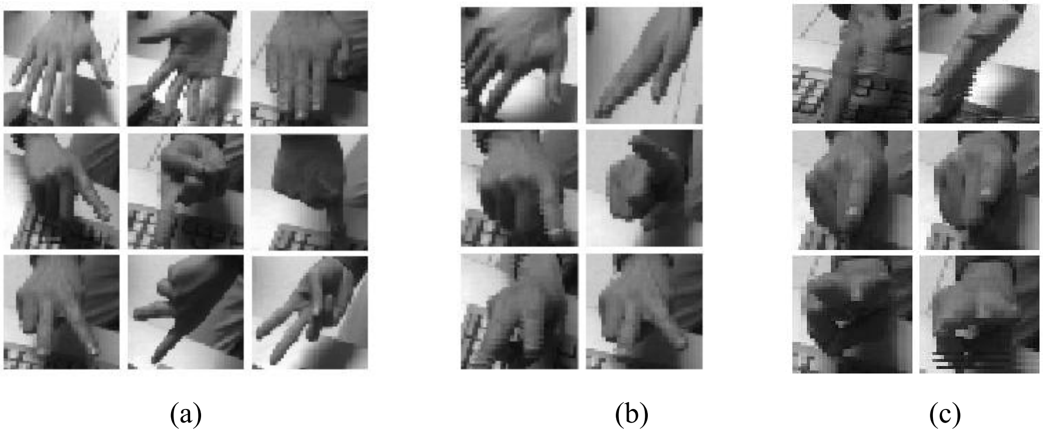


Figure 8. (a) Some correctly classified images by both DEM and KDEM (b) images that are mislabeled by DEM, but correctly labeled by KDEM (c) images that neither DEM nor KDEM can correctly label.

Table 1: Benchmark test: The average test error in percentage and its standard deviation

Error rate (%) and standard deviation		Benchmark		
		Banana	Breast-Cancer	Heart
Classification Method	RBF	10.8±0.06	27.6±0.47	17.6±0.33
	AdaBoost	12.3±0.07	30.4±0.47	20.3±0.34
	SVM	11.5±0.07	26.0±0.47	16.0±0.33
	KFD	10.8±0.05	25.8±0.48	16.1±0.34
	MDA	38.43±2.5	28.57±1.37	20.1±1.43
	KMDAñrandom	11.03±0.26	27.4±1.53	16.5±0.85
	KMDAñpca	10.7±0.25	27.5±0.47	16.5±0.32
	KMDAñevolutionary	10.8±0.56	26.3±0.48	16.1±0.33
	(# Kernel Vectors)	120	40	20

Table 2 Comparison of classification methods on Corel photos

Error rate (%) and standard deviation (Time used in second)	Training Dataset Size		
	40	60	80
MDA	15.82±1.13 (0.913 sec.)	11.26±0.82 (0.937 sec.)	8.68±0.27 (0.985 sec.)
BDA	11.24±0.97 (0.835 sec.)	10.03±0.89 (0.875 sec.)	7.59±0.11 (0.906 sec.)
DEM	13.37±0.93 (1.23 sec.)	9.51±0.51 (1.29 sec.)	7.88±0.32 (1.43 sec.)
KMDA	9.65±0.52 (1.101 sec.)	5.97±0.92 (1.954 sec.)	3.89±0.31 (1.981 sec.)
KBDA	9.36±0.72 (1.607 sec.)	8.13±0.79 (1.814 sec.)	6.82±0.18 (1.906 sec.)
KDEM	9.21±0.84 (1.927 sec.)	5.72±0.62 (2.363 sec.)	3.93±0.26 (2.552 sec.)

Table 3: Summary of the discriminant algorithms.

Issues	Comments
Problem Formulation	Well-known approaches • such as MDA; • EM steps in DEM, KDEM; • *Kernel trick* in all kernel algorithms.
Computational Efficiency	• Easy to implement, linear mapping such as MDA, BDA; • Though nonlinear projection, but kernel trick bypasses the computationally expensive non-linear projection, the computational cost is still affordable. Such as KMDA, KBDA; • Iterative procedure: DEM, KDEM.
Effectiveness	• Incorporation of unlabeled data in semi-supervised learning, DEM performs better than MDA; • With the introduction of *kernel trick*, nonlinear algorithms perform better than their linear algorithms; • KDEM works best for the combined reasons above.
Priori knowledge of Class Number	• Yes, MDA, DEM, KMDA, KDEM; • Without the prior knowledge of class number, MDA often reduces to two-class FDA, e.g., in content-based image retrieval (CBIR); • No, BDA, KBDA. (1+*x*)-classification problem
Training Data Size	• BDA works well for small training samples, e.g., in CBIR. • MDA, KMDA do not work well for very small training data; • The small sample problem is partially alleviated for DEM, KDEM with the introduction of unlabeled data.
Class Equivalence	• Yes. MDA, DEM, KMDA, KDEM; • No, biased towards one class such as BDA and KBDA.
Regularization	Yes for all discriminant algorithms.

Table 4: Error rate in percentage for KDEM on two-class synthetic data

Error Rate (%)		Correlation Coefficient of Positive Class				
		0	0.1	0.3	0.6	0.9
Correlation Coefficient of Negative Class	0	8.45	9.5	5.8	3.15	1.5
	0.1	9.5	7.9	6.85	4.05	1.2
	0.3	5.8	6.85	6.4	2.3	1.25
	0.6	3.15	4.05	2.3	1.6	0.35
	0.9	1.5	1.2	1.25	0.35	0

Table 5. View-independent hand posture recognition: Comparison among multi-layer perceptron (MLP), Nearest Neighbor (NN), Nearest Neighbor with growing templates (NN-G), EM, linear DEM and KDEM. The average error rate in percentage on 560 labeled and 14,000 unlabeled hand images with 14 different hand postures.

Algorithm	MLP	NN	NN-G	EM	DEM	KDEM
I-Feature	33.3	30.2	15.8	21.4	9.2	5.3
E-Feature	39.6	35.7	20.3	20.8	7.6	4.9