Towards Self-Exploring Discriminating Features

Ying Wu and Thomas S. Huang

Beckman Institute
University of Illinois at Urbana-Champaign
405 N. Mathews, Urbana, IL 61801
{yingwu,huang}@ifp.uiuc.edu
http://www.ifp.uiuc.edu/~yingwu

Abstract. Many visual learning tasks are usually confronted by some common difficulties. One of them is the lack of supervised information, due to the fact that labeling could be tedious, expensive or even impossible. Such scenario makes it challenging to learn object concepts from images. This problem could be alleviated by taking a hybrid of labeled and unlabeled training data for learning. Since the unlabeled data characterize the joint probability across different features, they could be used to boost weak classifiers by exploring discriminating features in a self-supervised fashion. Discriminant-EM (D-EM) attacks such problems by integrating discriminant analysis with the EM framework. Both linear and nonlinear methods are investigated in this paper. Based on kernel multiple discriminant analysis (KMDA), the nonlinear D-EM provides better ability to simplify the probabilistic structures of data distributions in a discrimination space. We also propose a novel data-sampling scheme for efficient learning of kernel discriminants. Our experimental results show that D-EM outperforms a variety of supervised and semi-supervised learning algorithms for many visual learning tasks, such as content-based image retrieval and invariant object recognition.

1 Introduction

Characterizing objects or concepts from images is one of the fundamental research topics of computer vision. Since there could be large variations in the image appearances due to various illumination conditions, viewing directions, variations in a general concept, this task is challenging because finding effective and explicit representations is generally a difficult problem. To approach this problem, machine learning techniques could be employed to model the variations in image appearances by learning the representations from a set of training data.

For example, invariant 3D object recognition is to recognize objects from different view directions. 3D object reconstruction suggests a way to invariantly characterize objects. Alternatively, objects could also be represented by their visual appearance without explicit reconstruction. However, representing objects 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, however, many other cases need to *learn* such features from a set of examples when image features are difficult to define. Many successful examples of learning approaches in the area of face and gesture recognition can be found in the literature [4, 2].

Generally, representing objects from examples requires huge training data sets, because input dimensionality is large and the variations that object classes undergo are significant. Although unsupervised or clustering schemes have been proposed [1, 20], it is difficult for pure unsupervised approaches to achieve accurate classification without supervision. Labels or supervised information of training samples are needed for recognition tasks. The generalization abilities of many current methods largely depend on training data sets. In general, good generalization requires large and representative labeled training data sets.

Unfortunately, collecting labeled data can be a tedious process. In some other cases, the situations are even worse, since it maybe impossible to label all the data. Content-based image retrieval is one of such examples.

The task of image retrieval is to find as many as possible "similar" images to the query images in a given database. Early research of image retrieval is searching by manually annotating every image in a database. To avoid manual annotating, an alternative approach is content-based image retrieval (CBIR), by which images would be indexed by their visual contents such as color, texture, shape, etc. Many research efforts have been made to extract these low-level image features [8, 15], evaluate distance metrics [13, 16], and look for efficient searching schemes [18]. However, it is generally impossible to find a fixed distance or similarity metrics. Such task could be cast as a classification problem, i.e., the retrieval system acts as a classifier to divide the images in the database into two classes, either relevant or irrelevant [22]. Unfortunately, one of the difficulties for learning is that only very limited number of query images could be used as labeled data, so that pure supervised learning with such limited training data can only give very weak classifiers.

We could consider the integration of pure supervised and unsupervised learning by taking hybrid data sets. The issue of combining unlabeled data in supervised learning begins to receive more and more research efforts recently and the research of this problem is still in its infancy. Without assuming parametric probabilistic models, several methods are based on the SVM [6, 3, 7]. However, when the size of unlabeled data becomes very large, these methods need formidable computational resources for mathematical programming. Some other alternative methods try to fit this problem into the EM framework and employ parametric models [22, 23], and have some applications in text classification [7, 11, 12]. Although EM offers a systematic approach to this problem, these methods largely depend on the *a priori* knowledge about the probabilistic structure of data distribution.

Since the labels of unlabeled data can be treated as missing values, The Expectation-Maximization (EM) approach can be applied to this problem. We assume that the hybrid data set is drawn from a mixture density distribution of C components $\{c_j, j = 1, \ldots, C\}$, which are parameterized by $\mathbf{\Theta} = \{\theta_j, j = 1, \ldots, C\}$. The mixture model can be represented as:

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{j=1}^{C} p(\mathbf{x}|c_j; \theta_j) p(c_j|\theta_j) \tag{1}$$

where $\mathbf{x}$ is a sample drawn from the hybrid data set $\mathcal{D} = \mathcal{L} \bigcup \mathcal{U}$. We make another assumption that each component in the mixture density corresponds to one class, i.e. $\{y_j = c_j, j = 1, \ldots, C\}$. Then, the joint probability density of the hybrid data set can be written as:

$$p(\mathcal{D}|\mathbf{\Theta}) = \prod_{\mathbf{x}_i \in \mathcal{U}} \sum_{j=1}^{C} p(c_j|\mathbf{\Theta}) p(\mathbf{x}_i|c_j; \mathbf{\Theta}) \bullet \prod_{\mathbf{x}_i \in \mathcal{L}} p(y_i = c_i|\mathbf{\Theta}) p(\mathbf{x}_i|y_i = c_i; \mathbf{\Theta})$$

The parameters $\mathbf{\Theta}$ can be estimated by maximizing *a posteriori* probability $p(\mathbf{\Theta}|\mathcal{D})$. Equivalently, this can be done by maximizing $\lg(p(\mathbf{\Theta}|\mathcal{D}))$. Let $l(\mathbf{\Theta}|\mathcal{D}) = \lg(p(\mathbf{\Theta})p(\mathcal{D}|\mathbf{\Theta}))$. A binary indicator $\mathbf{z}_i$ is introduced, $\mathbf{z}_i = (z_{i1}, \ldots, z_{iC})$. And $z_{ij} = 1$ iff $y_i = c_j$, and $z_{ij} = 0$ otherwise, so that

$$l(\mathbf{\Theta}|\mathcal{D}, \mathcal{Z}) = \lg(p(\mathbf{\Theta})) + \sum_{\mathbf{x}_i \in \mathcal{D}} \sum_{j=1}^{C} z_{ij} \lg(p(O_j|\mathbf{\Theta}) p(\mathbf{x}_i|O_j; \mathbf{\Theta})) \tag{2}$$

The EM algorithm can be used to estimate the parameters $\mathbf{\Theta}$ by an iterative hill climbing procedure, which alternatively calculates $E(\mathcal{Z})$, the expected values of all unlabeled data, and estimates the parameters $\mathbf{\Theta}$ given $E(\mathcal{Z})$. The EM algorithm generally reaches a local maximum of $l(\mathbf{\Theta}|\mathcal{D})$. It consists of two iterative steps:

- **E-step**: set $\hat{\mathcal{Z}}^{(k+1)} = E[\mathcal{Z}|\mathcal{D}; \hat{\mathit{\Theta}}^{(k)}]$
- **M-step**: set $\hat{\mathit{\Theta}}^{(k+1)} = \arg\max_{\theta} p(\mathit{\Theta}|\mathcal{D}; \hat{\mathcal{Z}}^{(k+1)})$

where $\hat{\mathcal{Z}}^{(k)}$ and $\hat{\mathit{\Theta}}^{(k)}$ denote the estimation for $\mathcal{Z}$ and $\mathit{\Theta}$ at the k-th iteration respectively. When the size of the labeled set is small, EM basically performs an unsupervised learning, except that labeled data are used to identify the components. If the probabilistic structure, such as the number of components in mixture models, is known, EM could estimate true parameters of the probabilistic model. Otherwise, the performance can be very bad. Generally, when we do not have such *a prior* knowledge about the data distribution, a Gaussian distribution is always assumed to represent a class. However, this assumption is often invalid in practice, which is partly the reason that unlabeled data hurt the classifier.

To alleviate such difficulties for the EM-based approaches, this paper proposes a novel approach, the *Discriminant-EM (D-EM)* algorithm, by inserting a step of discriminant analysis step into the EM iterations. Both linear and nonlinear discriminant analysis will be discussed in this paper. The proposed nonlinear method is based on kernel machines. A novel algorithm is presented for sampling training data for efficient learning of nonlinear kernel discriminants. We did standard benchmark testing of the kernel discriminant analysis. Our experiments of the D-EM algorithm include view-independent hand posture recognition and transductive content-based image retrieval.

2 Discriminant-EM Algorithm

As an extension to Expectation-Maximization, *Discriminant-EM (D-EM)* 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 D-EM algorithm loops between an expectation step, a discrimination step, and a maximization step. D-EM estimates the parameters of a generative model in a discrimination space.

The basic idea of this algorithm is to learn discriminating features and the classifier simultaneously by inserting a multi-class linear discrminant step in the standard expectation-maximization iteration loop. The basic idea of D-EM is to identify some "similar" samples in the unlabeled data set to enlarge the labeled data set so that supervised techniques are made possible in such an enlarged labeled set.

- **E-step**: set $\hat{\mathcal{Z}}^{(k+1)} = E[\mathcal{Z}|\mathcal{D}; \hat{\Theta}^{(k)}]$
- **D-step**: find a discriminant space and project data onto it
- **M-step**: set $\hat{\Theta}^{(k+1)} = \arg\max_\theta p(\Theta|\mathcal{D}; \hat{\mathcal{Z}}^{(k+1)})$

The E-step gives unlabeled data probabilistic labels, which are then used by the D-step to separate the data. D-EM makes assumption that the probabilistic structure of data distribution in the lower dimensional discrimination space is simplified and could be captured by lower order Gaussian mixtures. In this sense, the discriminant projection is not arbitrary. We will have a detailed discussion on the D-step in the next two sections, and concentrate on nonlinear discriminant analysis approaches.

D-EM begins with a weak classifier learned from the labeled set. Certainly, we do not expect much from this weak classifier. However, for each unlabeled sample $\mathbf{x}_j$, the classification confidence $\mathbf{w}_j = \{w_{jk}, k = 1, \ldots, C\}$ can be given based on the probabilistic label $\mathbf{l}_j = \{l_{jk}, k = 1, \ldots, C\}$ assigned by this weak classifier.

$$l_{jk} = \frac{p(\phi(\mathbf{x}_j)|c_k)p(c_k)}{\sum_{k=1}^{C} p(\phi(\mathbf{x}_j)|c_k)p(c_k)} \tag{3}$$

$$w_{jk} = -\lg(p(\phi(\mathbf{x}_j)|c_k)), \; k = 1, \ldots, C \tag{4}$$

Euqation(4) is just a heuristic to weight unlabeled data $\mathbf{x}_j \in \mathcal{U}$, although there may be many other choices.

After that, multiple discriminant analysis is performed on the new weighted data set,

$$\mathcal{D}' = \mathcal{L} \bigcup \{\mathbf{x}_j, \mathbf{l}_j, \mathbf{w}_j : \forall \mathbf{x}_j \in \mathcal{U}\},$$

by which the data set $\mathcal{D}'$ is projected to a new space of dimension $C-1$ but unchanging the labels and weights, i.e.,

$$\hat{\mathcal{D}} = \{\phi(\mathbf{x})_j, y_j : \forall \mathbf{x}_j \in \mathcal{L}\} \bigcup \{\phi(\mathbf{x})_j, \mathbf{l}_j, \mathbf{w}_j : \forall \mathbf{x}_j \in \mathcal{U}\}. \tag{5}$$

Then parameters $\mathbf{\Theta}$ of the probabilistic models are estimated by maximizing a posteriori probability on $\hat{\mathcal{D}}$, so that the probabilistic labels are given by the Bayesian classifier according to Equation(3). The D-EM algorithm iterates over these three steps, "Expectation-Discrimination-Maximization".

3 Linear Multiple Discriminant Analysis

Multiple discriminant analysis (MDA) is a natural generalization of Fisher's linear discriminant analysis (LDA) for the case of multiple classes [5]. The goal of MDA is to find a linear projection $\mathbf{W}$ that maps the original d_1-dimensional data space $\mathcal{X}$ to a d_2-dimensional discrimination space Δ ($d_2 \leq c-1$, c is the number of classes) such that the classes are linearly separable.

More specifically, MDA finds the best linear projection of labeled data, $\mathbf{x} \in \mathcal{X}$, such that the ratio of between-class scatter, S_B, to within-class scatter, S_W, is maximized. Let n be the size of training data set, and n_j be the size of the data set for class j. Then,

$$\mathbf{V}_{opt} = \arg\max_{\mathbf{V}} \frac{|\mathbf{V}^T S_B \mathbf{V}|}{|\mathbf{V}^T S_W \mathbf{V}|} \tag{6}$$

$$S_B = \sum_{j=1}^{c} n_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T, \tag{7}$$

$$S_W = \sum_{j=1}^{c} \sum_{k=1}^{n_j} (\mathbf{x}_k - \mathbf{m}_j)(\mathbf{x}_k - \mathbf{m}_j)^T, \tag{8}$$

where the total mean and class means are given by

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k,$$
$$\mathbf{m}_j = \frac{1}{n_j} \sum_{k=1}^{n_j} \mathbf{x}_k, \ \forall j \in \{1, \ldots, c\}$$

and $\mathbf{V}_{opt} = [\mathbf{v}_1, \ldots, \mathbf{v}_{c-1}]$ will contain in its columns $c-1$ eigenvectors corresponding to $c-1$ eigenvalues, i.e.,

$$S_B \mathbf{v}_i = \lambda_i S_W \mathbf{v}_i.$$

4 Nonlinear Discriminant Analysis

Nonlinear discriminant analysis could be achieved by transforming the original data space $\mathcal{X}$ to a nonlinear feature space $\mathcal{F}$ and then performing LDA in $\mathcal{F}$. This section presents a kernel-based approach.

4.1 Kernel Discriminant Analysis

In *nonlinear* discriminant analysis, we seek a prior transformation of the data, $\mathbf{y} = \phi(\mathbf{x})$, that maps the original data space $\mathcal{X}$, to a feature space (F-space) $\mathcal{F}$, in which MDA can be then performed. Thus, we have

$$\mathbf{V}_{opt} = \arg\max_{\mathbf{V}} \frac{|\mathbf{V}^T S_B^\phi \mathbf{V}|}{|\mathbf{V}^T S_W^\phi \mathbf{V}|}, \tag{9}$$

$$S_B^\phi = \sum_{j=1}^{c} n_j(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T, \tag{10}$$

$$S_W^\phi = \sum_{j=1}^{c}\sum_{k=1}^{n_j}(\phi(\mathbf{x}_k) - \mathbf{m}_j)(\phi(\mathbf{x}_k) - \mathbf{m}_j)^T, \tag{11}$$

with

$$\mathbf{m} = \frac{1}{n}\sum_{k=1}^{n}\phi(\mathbf{x}_k),$$

$$\mathbf{m}_j = \frac{1}{n_j}\sum_{k=1}^{n_j}\phi(\mathbf{x}_k),\ \forall j \in \{1,\ldots,c\}.$$

In general, because we choose $\phi(\cdot)$ to facilitate *linear* discriminant analysis in the feature space $\mathcal{F}$, the dimension of the feature space may be arbitrarily large, even infinite. As a result, the explicit computation of the mapping induced by $\phi(\cdot)$ could be prohibitively expensive.

The problem can be made tractable by taking a kernel approach that has recently been used to construct nonlinear versions of support vector machines [19], principal components analysis [17], and invariant feature extraction [9, 14]. Specifically, the observation behind kernel approaches is that if an algorithm can be written in such a way that only dot products of the transformed data in $\mathcal{F}$ need to be computed, explicit mappings of individual data from $\mathcal{X}$ become unnecessary.

Referring to Equation 9, we know that any column of the solution $\mathbf{V}$, must lie in the span of all training samples in $\mathcal{F}$, i.e., $\mathbf{v}_i \in \mathcal{F}$. Thus, for some $\underline{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$,

$$\mathbf{v} = \sum_{k=1}^{n}\alpha_k\phi(\mathbf{x}_k) = \Phi\underline{\alpha}, \tag{12}$$

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 $\mathcal{F}$ as follows (we will drop the subscript on $\mathbf{v}_i$ in the ensuing):

$$\mathbf{v}^T \phi(\mathbf{x}_k) = \underline{\alpha}^T \Phi^T \phi(\mathbf{x}_k) = \underline{\alpha}^T \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix} = \underline{\alpha}^T \xi_k, \tag{13}$$

where

$$\xi_k = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix}, \tag{14}$$

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

$$\mathbf{v}^T \mathbf{m}_j = \underline{\alpha}^T \frac{1}{n_j} \sum_{k=1}^{n_j} \begin{bmatrix} \phi^T(\mathbf{x}_1)\phi(\mathbf{x}_k) \\ \vdots \\ \phi^T(\mathbf{x}_n)\phi(\mathbf{x}_k) \end{bmatrix} \tag{15}$$

$$= \underline{\alpha}^T \begin{bmatrix} \frac{1}{n_j} \sum_{k=1}^{n_j} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ \frac{1}{n_j} \sum_{k=1}^{n_j} k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix} = \underline{\alpha}^T \mu_j. \tag{16}$$

It follows that

$$\mathbf{v}^T S_B \mathbf{v} = \underline{\alpha}^T K_B \underline{\alpha}, \tag{17}$$

where

$$K_B = \sum_{j=1}^{c} n_j (\mu_j - \mu)(\mu_j - \mu)^T, \tag{18}$$

and

$$\mathbf{v}^T S_W \mathbf{v} = \underline{\alpha}^T K_W \underline{\alpha}, \tag{19}$$

where

$$K_W = \sum_{j=1}^{c} \sum_{k=1}^{n_j} (\xi_k - \mu_j)(\xi_k - \mu_j)^T. \tag{20}$$

The goal of Kernel Multiple Discriminant Analysis (KMDA), then, is to find

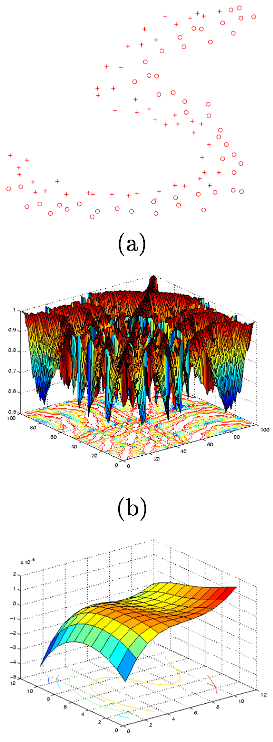
$$\mathbf{A}_{opt} = \arg\max_{\mathbf{A}} \frac{|\mathbf{A}^T K_B \mathbf{A}|}{|\mathbf{A}^T K_W \mathbf{A}|}, \tag{21}$$

where $\mathbf{A} = [\underline{\alpha}_1, \cdots, \underline{\alpha}_{c-1}]$, and computation of K_B and K_W requires only kernel computations.

4.2 Sampling Data for Efficiency

Because K_B and K_W are $n \times n$ matrices, where n is the size of training set, the nonlinear mapping is dependent on the entire training samples. For large n, the solution to the generalized eigensystem is costly. Approximate solutions could be obtained by sampling representative subsets of the training data, $\{p_k | k = 1, \ldots, M, M < n\}$, and using $\tilde{\xi}_k = [k(\mathbf{x}_1, \mathbf{x}_k), \cdots, k(\mathbf{x}_M, \mathbf{x}_k)]^t$ to take the place of ξ_k.

We select representatives, or *kernel vectors*, by identifying those training samples which are likely to play a key role in $\Xi = [\xi_1, \ldots, \xi_n]$. Ξ is an $n \times n$ matrix, but $rank(\Xi) \ll n$, when the size of training data set is very large. This fact suggests that some training samples could be ignored in calculating kernel features ξ.



(a)

(b)

(c)

Fig. 1. KMDA with a 2D 2-class non-linearly-separable example. (a) Original data (b) the kernel features of the data (c) the nonlinear mapping.

Our approach 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 $\mathbf{y}_i^{(k)} = \mathbf{V}^{(k)T}\phi(\mathbf{x}_i) = \mathbf{A}_{opt}^{(k)T}\xi_I^{(k)} \in \Delta$ 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., $e^{(k)} \geq e^{(k-1)}$, and we terminate. The evolutionary kernel vector selection algorithm is summarized below in Figure 2.

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

// Initialization
$k = 0$; $e = \infty$; $KV^{(0)}$ =random_pick($X$);
do{
    // Perfrom KMDA
    $\mathbf{A}_{opt}^{(k)}$ =KMDA($\underline{X}$, $KV^{(k)}$);
    // Project $\mathcal{X}$ to $\Delta$
    $Y^{(k)}$ =Proj($X$, $\mathbf{A}_{opt}^{(k)}$);

    //Bayesian classifier
    $\Theta^{(k)}$ =Bayes($Y^{(k)}$, $L$);
    // Classification
    $\bar{L}^{(k)}$ =Labeling($Y^{(k)}$, $\Theta^{(k)}$);
    // Calculate error
    $e^{(k)}$ =Error($\bar{L}^{(k)}$, $L$);

    // Select new kernel vectors
    if($e^{(k)} < e$)
        $e = e^{(k)}$; $KV = KV^{(k)}$; $k++$;
        $KV^{(k)}$ =random_pick($\{\mathbf{x}_i : \bar{l}_i^{(k)} \neq l_i\}$);
    else
        $KV = KV^{(k-1)}$; break;
    end
}
return $KV$;
```

Fig. 2. Evolutionary Kernel Vector Selection

4.3 Kernel D-EM Algorithm

We now apply KMDA to D-EM. *Kernel D-EM (KDEM)* is a generalization of linear D-EM, in which instead of a simple linear transformation of the data, KMDA is used to project the data nonlinearly into a feature space where the data is better separated linearly. 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 $\mathcal{X}$ to the discrimination space Δ, which is a linear subspace of the feature space $\mathcal{F}$, is given by $\mathbf{V}^T\phi(\cdot)$ implicitly or $\mathbf{A}^T\xi$ explicitly. A low-dimensional generative model is used to capture the transformed data in Δ.

Empirical observations suggest that the transformed data often 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 D-EM can be initialized by selecting all labeled data as kernel vectors, and training a weak classifier based on only unlabeled samples.

5 Experiments

In this section, we compare KMDA with other supervised learning techniques on some standard data sets. Experimental results of D-EM on content-based image retrieval and view-independent hand posture recognition are presented.

5.1 Benchmark Test for KMDA

We first verify the ability of KMDA with our data-sampling algorithms. Several benchmark data sets[1] are used in our experiments. The benchmark data has 100 different realizations. In [9], results of different approaches on these data sets have been reported. The proposed KMDA algorithms were compared to a single RBF classifier (RBF), a support vector machine (SVM), AdaBoost, and the kernel Fisher discriminant (KFD) [10]. RBF kernels were used in all kernel-based algorithms.

In Table 1, KMDA-pca is KMDA with PCA selection, and KMDA-evol is KMDA with evolutionary selection, where #-KVs is the number of kernel vectors. The benchmark tests show that the proposed approaches achieve comparable results as other state-of-the-art techniques, in spite of the use of a decimated training set.

5.2 Content-based Image Retrieval

Using a random subset of the database or even the whole database as an unlabeled data set, the D-EM algorithm identifies some "similar" images to the labeled images to enlarge the labeled data set. Therefore, good discriminating

[1] The standard benchmark data sets in our experiments are obtained from `http://www.first.gmd.de/~raetsch`.

Table 1. Benchmark Test: the average test error as well as standard deviation.

Benchmark	Banana	B-Cancer	Heart	Thyroid	F-Sonar
RBF	10.8±0.06	27.6±0.47	17.6±0.33	4.5±0.21	34.4±0.20
AdaBoost	12.3±0.07	30.4±0.47	20.3±0.34	4.4±0.22	35.7±0.18
SVM	11.5±0.07	26.0±0.47	16.0±0.33	4.8±0.22	32.4±0.18
KFD	10.8±0.05	25.8±0.46	16.1±0.34	4.2±0.21	33.2±0.17
KMDA-evol	10.8±0.56	26.3±0.48	16.1±0.33	4.3±0.25	33.3±0.17
#-KVs	120	40	20	20	40

features could be automatically selected through this enlarged training data set to better represent the implicit concepts. The application of D-EM to image retrieval is straightforward. In our current implementation, in the transformed space, both classes are represented by a Gaussian distribution with three parameters, the mean μ_i, the covariance $\mathbf{\Sigma}_i$ and *a priori* probability of each class P_i. The D-EM iteration tries to boost an initial weak classifier.

In order to give some analysis and compare several different methods, we manually label an image database of 134 images, which is a subset of the COREL database. All images in the database have been labeled by their categories. In all the experiments, these labels for unlabeled data are only used to calculate classification error.

To investigate the effect of the unlabeled data used in D-EM, we feed the algorithm a different number of labeled and unlabeled samples. The labeled images are obtained by relevance feedback. When using more than 100 unlabeled samples, the error rates drop to less than 10%. From Figure 3, we find that D-EM brings about 20% to 30% more accuracy. In general, combining some unlabeled data can largely reduce the classification error when labeled data are very few.

Our algorithm is also tested by several large databases. The COREL database contains more than 70, 000 images over a wide range of more than 500 categories with 120 × 80 resolution. The VISTEX database is a collection of 832 texture images. Satisfactory results are obtained.

5.3 View-independent Hand Posture Recognition

Next, we examine results for KDEM on a hand gesture recognition task. The task is to classify among 14 different hand postures, each of which represents a gesture command mode, such as navigating, pointing, grasping, etc. Our raw data set 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[21]. Gabor wavelet filters with 3 levels and 4 orientations are used to extract 12 texture features.

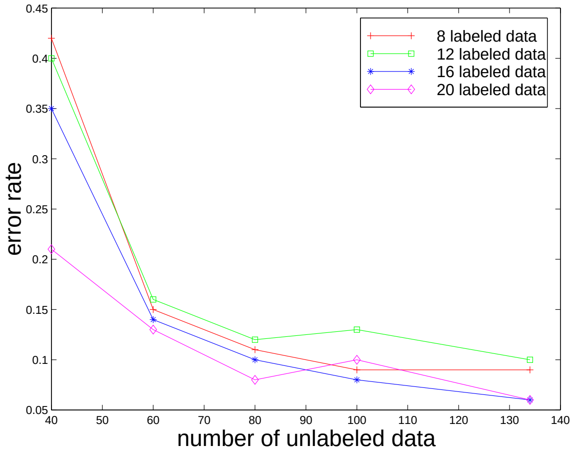


Fig. 3. The effect of labeled and unlabeled data in D-EM. Error rate decreases when adding more unlabeled data. Combining some unlabeled data can largely reduce the classification error.

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 also represent images by coefficients of the 22 largest principal components of the total data set resized to 20×20 pixels (these are “eigenimages”, or E-Features) [21]. In our experiments, we use 140 (10 for each) and 10000 (randomly selected from the whole database) labeled and unlabeled images respectively, for training with both EM and D-EM. Table 2 shows the comparison.

Table 2. View-independent hand posture recognition: Comparison among multilayer perceptron (MLP),Nearest Neighbor with growing templates (NN-G), EM, linear D-EM (LDEM) and KDEM

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

We observed that multilayer perceptrons are often trapped in local minima and nearest neighbor 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 LDEM and KDEM outperform other methods, since the D-step optimizes separability of the classes. Finally, note the effectiveness of KDEM. We find that KDEM often appears to project classes to approximately Gaussian clusters in the transformed space, which facilitates their modeling with Gaussians.

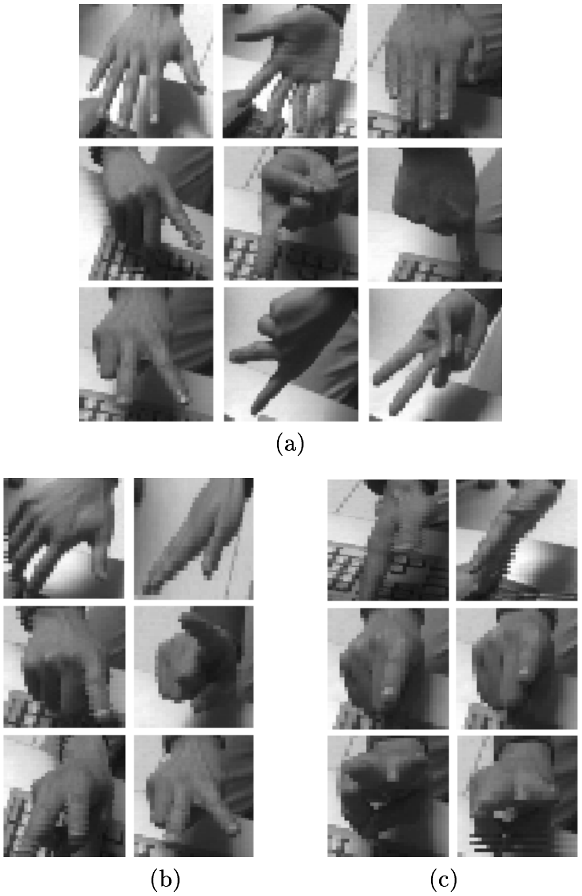


Fig. 4. (a) Some correctly classified images by both LDEM and KDEM (b) images that are mislabeled by LDEM, but correctly labeled by KDEM (c) images that neither LDEM or KDEM can correctly labeled.

6 Conclusion and Future Work

Many visual learning tasks are confronted by some common difficulties, such as the lack of a large number of supervised training data, and learning in high dimensional space. In this paper, we presented a self-supervised learning technique, Discriminant-EM, which employs both labeled and unlabeled data in training, and explores most discriminant features automatically. Both linear and nonlinear approaches were investigated. We also presented a novel algorithm for efficient kernel-based, nonlinear, multiple discriminant analysis (KMDA). The algorithm identifies "kernel vectors" which are the defining training data for the purposes of classification. Benchmark tests show that KMDA with these adaptations performs comparably with the best known supervised learning algorithms. On real experiments for recognizing hand postures and content-based image retrieval, D-EM outperforms naïve supervised learning and existing semi-supervised algorithms.

Examination of the experimental results reveals that KMDA often maps data sets corresponding to each class into approximately Gaussian clusters in the tranformed space, even when the initial data distribution is highly non-Gaussian. In future work, we will investigate this phenomenon more closely.

Acknowledgments

This work was supported in part by National Science Foundation Grants CDA-96-24396 and IRI-96-34618 and NSF Alliance Program.

References

1. R. Basri, D. Roth, and D. Jacobs. Clustering appearances of 3D objects. In *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, pages 414–420, 1998.
2. P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *Proc. of European Conference on Computer Vision*, April 1996.
3. K. Bennett. Combining support vector and mathematical programming methods for classification. In *Advances in Kernel Methods - Support Vector Learning.* Cambridge, MA: MIT Press, 1999.
4. Y. Cui and J. Weng. Hand segmentation using learning-based prediction and verification for hand sign recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 88–93, 1996.
5. R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.
6. A. Gammerman, V. Vapnik, and V. Vowk. Learning by transduction. In *Proc. of Conf. Uncertainty in Artificial Intelligence*, pages 148–156, 1998.
7. T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of Int'l Conf. on Machine Learning*, pages 200–209, 1999.
8. B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Ananlysis and Machine Intelligence*, 18:837–841, Nov. 1996.
9. Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, Alex Smola, and Klaus-Robert Müller. Invariant feature extraction and classification in kernel spaces. In S. Solla, T. Leen, and K. Muller, editors, *Advances in Neural Information Processing Systems 12*, pages 526–532, Cambridge, MA, 2000. MIT Press.

10. Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, Alex Smola, and Klaus-Robert Müller. Fisher discriminant analysis with kernels. In *IEEE workshop on Neural Networks for Signal Proceesing*, 1999.
11. Tom Mitchell. The role of unlabeled data in supervised learning. In *Proc. Sixth Int'l Colloquium on Cognitive Science*, Spain, 1999.
12. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 1999.
13. M. Popescu and P. Gader. Image content retrieval from image database using feature integration by choquet integral. In *Proc. SPIE Storage and Retrieval for Image and Video Database*, volume VII, 1998.
14. Volker Roth and Volker Steinhage. Nonlinear discriminant analysis using kernel functions. In S. Solla, T. Leen, and K. Muller, editors, *Advances in Neural Information Processing Systems*, pages 568–574, Cambridge, MA, 2000. MIT Press.
15. Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 8:644–655, 1998.
16. S. Santini and R. Jain. Similarity measures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:871–883, 1999.
17. Bernhard Schölkopf, Alexander Smola, and Klaus Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
18. D. Swets and J. Weng. Hierarchical discriminant analysis for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:386–400, 1999.
19. V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.
20. M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 101–108, Hilton Head Island, South Carolina, 2000.
21. Ying Wu and Thomas S. Huang. View–independent recognition of hand postures. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 88–94, Hilton Head Island, South Carolina, June 2000.
22. Ying Wu, Qi Tian, and Thomas S. Huang. Discriminant-EM algorithm with application to image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 222–227, Hilton Head Island, South Carolina, June 2000.
23. Ying Wu, Kentaro Toyama, and Thomas S. Huang. Self-supervised learning for object recognition based on kernel Discriminant-EM algorithm. In *Proc. IEEE Int'l Conference on Computer Vision*, volume I, pages 275–280, Vancouver, July 2001.