Towards Self-Exploring Discriminating Features for Visual Learning

Ying Wu [1]

Electrical & Computer Engineering, Northwestern University,

2145 Sheridan Road, Evanston, IL 60208

Thomas S. Huang

Beckman Institute, University of Illinois at Urbana-Champaign,

405 N. Mathews, Urbana, IL 61801

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. Another difficulty is the high dimensionality of the visual data. Fortunately, these difficulties could be alleviated by using 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. This paper proposes a novel method, the Discriminant-EM (D-EM) algorithm, which attacks these difficulties by integrating discriminant analysis with the EM framework in this hybrid formulation. 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, invariant object recognition, and nonstationary color tracking. The proposed approach could be easily applied for many other learning tasks.

Key words: Visual learning, Unlabeled data, Discriminant-EM, Object recognition, Content-based image retrieval, nonstationary color tracking

1 Introduction

Characterizing objects or concepts from visual data 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 and conceptual ambiguities, this task is challenging because finding effective and explicit representations is in general a difficult problem. To approach to this problem, machine learning techniques can 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. Full 3D reconstruction of the target suggests a way of

Email addresses: yingwu@ece.northwestern.edu (Ying Wu), huang@ifp.uiuc.edu (Thomas S. Huang).

URLs: http://www.ece.nwu.edu/~yingwu (Ying Wu), http://www.ifp.uiuc.edu (Thomas S. Huang).

[1] Corresponding author. Tel.:+1-847-4912901; Fax: +1-847-4914455.

invariant target representation. Alternatively, without explicit reconstruction, objects could also be represented by their visual appearances. 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, in many other cases, such features have to be *learned* from a set of examples when they are difficult to be specified. Many successful examples of learning approaches in the area of face and gesture recognition can be found in the literature (Cui and Weng, 1996; Belhumeur et al., 1996).

In general, representing objects by visual examples requires huge training data sets, because the data dimensionality is very high and the variations that object classes undergo are significant. Although unsupervised or clustering schemes have been proposed (Basri et al., 1998; Weber et al., 2000), it is difficult for pure unsupervised approaches to achieve accurate classification without supervision. Annotations 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 annotated or labeled training data sets.

Unfortunately, collecting labeled data can be a tedious process. In some 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 approaches of image retrieval were based on the keyword search on the image databases. Those keyword annotations were made manually in advance. Obviously, it is in general impossible to use a finite

set of keywords to describe or to represent an image. In addition, manually annotating a large image database is painstaking.

To avoid manually keyword annotating, an alternative approach is called 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 (Manjunath and Ma, 1996; Rui et al., 1998), evaluate distance metrics (Popescu and Gader, 1998; Santini and Jain, 1999), and look for efficient searching schemes (Swets and Weng, 1999). However, it is generally impossible to find fixed distance measurements or similarity metrics to measure the similarities of different images based on these image features. In another perspective, the retrieval 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 (Wu et al., 2000). Unfortunately, one of the difficulties for this learning approach is that only a very limited number of query images could be used as labeled training data, so that pure supervised learning with such limited training data can only give very weak classifiers. Fortunately, unlabeled data might be combined with labeled data to facilitate a possible successful learning.

Besides invariant 3D object recognition and content-based image retrieval, more interestingly, model transduction learning is highly related to (and even can be formulated by) such a hybrid learning problem based on both labeled and unlabeled data. Model transduction is to adapt an old model to a set of unsupervised new data to produce a new model. For example, in the case of adaptive color tracking, color model at time t could be adapted to new color images at $t + 1$ for better color segmentation, since the lighting at time

$t+1$ could be different from that at t. It is a good practice to learn a generic color classifier by collecting a large labeled data set (Jones and Rehg, 1998). If some color invariants to lighting could be found, learning such a color classifier would suggest a direct and robust way to color tracking. However, when we consider the non-stationary nature of color distributions over time, we do not generally expect to find such invariants.

The approach taken in (Jones and Rehg, 1998) is an *inductive learning* approach, by which the color classifier learned should be able to classify any pixel in any image. Obviously, this color classifier would be highly nonlinear, and a huge labeled training data set is required to achieve good generalization. However, in the color tracking scenario, the requirement of generalization could be relaxed to a subset of the data space, e.g., a specific image. Specifically, a color classifier M_t at time frame t could be only used to classify pixel $\mathbf{x}_j$ in the current specific image feature data set I_t so that this specific classifier M_t could be simpler than a generic classifier. When there is a new image I_{t+1} at time $t+1$, this specific classifier M_t should be *transduced* to a new classifier M_{t+1} which works just for the new image I_{t+1} instead of I_t. The classification can be described as:

$$y_i = arg \max_{j=1,\ldots,C} p(y_j|\mathbf{x}_i, M_t, I_{t+1} : \forall \mathbf{x}_i \in I_{t+1}) \tag{1}$$

where y_i is the label of $\mathbf{x}_i$, and C is the number of classes. In this sense, we do not care the performance of the classifier M_{t+1} outside I_{t+1}. The *transductive learning* is to transduce the classifier M_t to M_{t+1} given I_{t+1}. Figure 1 shows the transduction of color classifiers.

This *transduction* may not always be feasible unless we know the joint distribu-

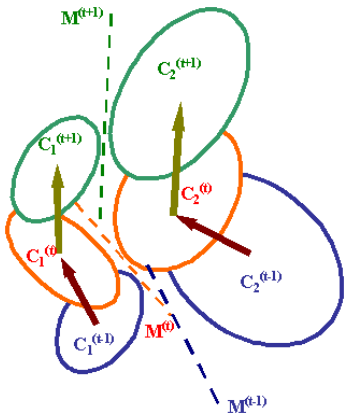


Fig. 1. An illustration of transduction of classifiers.

tion of I_t and I_{t+1}. Unfortunately, such joint probability is generally unknown since we may not have enough *a priori* knowledge about the transition in a color space over time. One approach is to assume a transition model so that we can explicitly model $p(I_{t+1}|I_t)$. One of the difficulties of this approach is that a fixed transition model is lack of flexibility and is unable to capture much dynamics. The approach used in (Raja et al., 1998) assumes a linear transition model. However, the transition (updating) of color models is plagued since the newest image has not been segmented yet.

However, different from the transition model assumption, we assume that the classifier M_t at time t can give "confident" labels to several samples in I_{t+1}, so that the data in I_{t+1} can be divided into two parts: labeled data set $\mathcal{L} = \{(\mathbf{x}_j, y_j), j = 1, \ldots, N\}$, and unlabeled set $\mathcal{U} = \{\mathbf{x}_j, j = 1, \ldots, M\}$, where N and M are the size of the labeled set and unlabeled set respectively, $\mathbf{x}_j$ is the color feature vector of a pixel in this case, and y_j is its label (such as skin tone or non-skin tone). Here, $\mathcal{L}$ and $\mathcal{U}$ are from the same distribution. Consequently, the transductive classification can be written as:

$$y_i = arg \max_{j=1,\ldots,C} p(y_j|\mathbf{x}_i, \mathcal{L}, \mathcal{U} : \forall \mathbf{x}_i \in \mathcal{U}) \tag{2}$$

In this formulation, the specific classifier M_t is transduced to another classifier M_{t+1} by combining a large unlabeled data set from I_{t+1}.

From the formulation of above three different visual learning tasks, the new learning paradigm integrates pure supervised and unsupervised learning by taking hybrid data sets. The issue of combining unlabeled data in supervised learning has begun to receive more and more research efforts recently and the research of this problem is still in its infancy.

Without assuming parametric probabilistic models for the data distributions, several methods were proposed based on the Support Vector Machines (SVM) (Gammerman et al., 1998; Bennett, 1999; Joachims, 1999). However, when the size of unlabeled data becomes very large, these methods need formidable computational resources for mathematical programming involved.

Another very interesting approach to learning from hybrid data is to exploit the cross-modality structure of the training data (de Sa, 1994; de Sa and Ballard, 1998). For some of the unlabeled samples, their labels can be obtained by making use of the structure of the feature distributions over different sensory modalities. Taking a similar idea, the method in (Blum and Mitchell, 1998; Mitchell, 1999; Riloff and Jones, 1999; Wu and Huang, 2001) described the co-training approach to Web page classification. The basic idea is to train two independent classifiers rather than one. Initially, two classifiers are trained using whatever labeled training examples available. This results in two imperfect classifiers. Each classifier is allowed to examine the unlabeled data and to pick its most confidently predicted positive and negative examples, and add them to the set of labeled examples. Then, both classifiers are now retrained on this augmented set of labeled data set, and the process is repeated until it con-

verges. However, this algorithm is based on an assumption that the features to train two classifiers are *redundantly sufficient* to the classification problem. Blum and Mitchell (Blum and Mitchell, 1998; Mitchell, 1999) applied it to Web page classification. Riloff and Jones (Riloff and Jones, 1999) employed this approach to learn to classify noun phrase as positive and negative examples of locations. De Sa and Ballard (de Sa and Ballard, 1998) employed this approach to classify speech phonemes based on both the audio signal and the video signal watching the speaker's lips. Wu and Huang (Wu and Huang, 2001) extended this idea to model transduction for multimodal visual tracking.

Besides SVM-based approaches and co-learning based approaches, an alternative method fits this problem into the EM framework and employs parametric probabilistic models (Wu et al., 2000, 2001). The basic EM scheme seems a solution to this problem (Ghahramani and Jordan, 1994; Tresp et al., 1995), since the labels of unlabeled data can be treated as missing values, which can be estimated by the EM algorithm. Combing a set of unlabeled data in training, classification accuracy can be improved by the EM algorithm. Some successful applications of this approach include text classification (Joachims, 1999; Mitchell, 1999; Nigam et al., 1999).

However, several assumptions have to be made for this approach. The first assumption is that data are generated by a mixture model. Another assumption is that there is a correspondence between mixture components and classes. Since there may be a discrepancy between the generative model and the ground truth data distribution, this approach would fail. Although EM offers a systematic approach to this problem, these methods largely depend on the *a priori* knowledge about the probabilistic structures of data distributions. It poses some difficulties when we use parametric generative models. We should

develop more robust methods when the probabilistic structure of true data distribution disagrees with the structure of the generative model. In addition, we should be able to handle the learning in a high-dimensional space, for example, learning visual data.

In this paper, a novel method, called Discriminant-EM, is proposed to approach to the learning tasks on hybrid training data sets, by integrating supervised and unsupervised learning paradigms and identifying most discriminating features in a self-supervised fashion. Section 2 will describe an approach based on the Expectation-Maximization framework. When revealing some difficulties for this approach for many visual learning problems, we propose a new algorithm, the Discriminant-EM algorithm. Both linear D-EM algorithm and kernel-based nonlinear D-EM algorithm will be described in Section 3 and Section 4, respectively. Extensive experiments on the above three computer vision applications, including content-based image retrieval, view invariant object recognition, and nonstationary color tracking, will be presented at Section 5.

2 An Expectation-Maximization Approach

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

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^{C} p(\mathbf{x}|c_j; \theta_j) p(c_j|\theta_j) \tag{3}$$

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:

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

The parameters Θ can be estimated by maximizing *a posteriori* probability $p(\Theta|\mathcal{D})$. Equivalently, this can be done by maximizing $\lg(p(\Theta|\mathcal{D}))$. Let $l(\Theta|\mathcal{D}) = \lg(p(\Theta)p(\mathcal{D}|\Theta))$. 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

$$l(\Theta|\mathcal{D}, \mathcal{Z}) = \lg(p(\Theta)) + \sum_{\mathbf{x}_i \in \mathcal{D}} \sum_{j=1}^{C} z_{ij} \lg(p(c_j|\Theta) p(\mathbf{x}_i|c_j;\Theta)) \tag{4}$$

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

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

where $\hat{\mathcal{Z}}^{(k)}$ and $\hat{\Theta}^{(k)}$ denote the estimation for $\mathcal{Z}$ and Θ 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.

The EM algorithm is a general and solid approach to deal with hidden variables. To model the data distributions, parametric generative models are often employed, since they are analyzable and flexible. The Gaussian Mixture model is a frequent choice. Although parametric generative models offer good analytical properties, they also bring some problems. In practice, especially in vision problems, learning techniques are performed in high-dimensional spaces. Consequently, the dimensionality of the generative model will be also very high, such that the M-step has to estimate numerous model parameters. If the training data set is not large enough, the estimation could be highly biased and numerically unstable. Although some regularization approaches have been proposed to handle such circumstances, we will still ask whether it is necessary to perform learning in such a high-dimensional space? Is it possible to reduce the dimensionality in learning?

To alleviate these difficulties for the EM-based approaches, this paper proposes a novel approach, the *Discriminant-EM (D-EM)* algorithm, by inserting a 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 have performed standard benchmark testing of the kernel discriminant analysis. Our experiments of the D-EM algorithm include view-independent hand posture recognition, transductive content-based image retrieval, and nonsta-

tionary color tracking.

3 The Linear D-EM Algorithm

Since we generally do not know the probabilistic structure of a data distribution, e.g., the number of mixed Gaussian components, the EM algorithm often fails when structure assumption of the generative model does not hold. One approach to this problem is to try every possible structure and select the best one. However, this requires more computational resources. An alternative is to find a mapping such that the data are clustered in the mapped data space, in which the probabilistic structure could be simplified and captured by simpler Gaussian mixture models. The multiple discriminant analysis (MDA) technique offers a way to relax the assumption of probabilistic structure. The basic idea of our approach is to learn a mapping as well as the generative model parameters by inserting MDA into the EM iterations, in which EM will provide MDA a large labeled data set to select most discriminating features.

3.1 Linear multiple discriminant analysis

Multiple discriminant analysis (MDA) (Duda and Hart, 1973) is a natural generalization of Fisher's linear discrimination (LDA) in the case of multiple classes. MDA offers many advantages and has been successfully applied to many tasks such as face recognition. The basic idea behind MDA is to find a linear transformation $\mathbf{W}$ to map the original d_1-dimensional data space to a new d_2 space such that the ratio between the between-class scatter and

within-class scatter is maximized in the new space.

$$\mathbf{W} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T S_b \mathbf{W}|}{|\mathbf{W}^T S_w \mathbf{W}|} \tag{5}$$

Suppose $\mathbf{x}$ is an m-dimensional random vector drawn from C classes in the original data space. The ith class has a prior probability P_i, a mean vector $\mathbf{m}_i$. The within-class scatter matrix S_w is defined by

$$S_w = \sum_{i=1}^{C} E[(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T | c_i] \tag{6}$$

where c_i denotes the ith class. The between-class scatter matrix S_b defined by

$$S_b = \sum_{i=1}^{C} P_i \cdot (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \tag{7}$$

where the grand mean $\mathbf{m}$ is defined as $\mathbf{m} = E[\mathbf{x}] = \sum_{i=1}^{C} P_i \mathbf{m}_i$. Details can be found in (Duda and Hart, 1973).

MDA provides a means to catch the major differences between classes and discount factors that are not related to classification. Some features most relevant to classification are automatically selected or combined by the linear mapping $\mathbf{W}$ in MDA, although these features may not have substantial physical meanings any more. Another advantage of MDA is that the data are clustered to some extent in the projected space, which makes it easier to select the structure of Gaussian mixture models.

It is apparent that MDA is a supervised statistical method, which requires large labeled training sets to estimate some statistics such as mean and covariance. By combining MDA with the EM framework, our proposed method, the Discriminant-EM algorithm (D-EM), is such a way that combines supervised and unsupervised paradigms. The basic idea of D-EM is to enlarge the

labeled data set by identifying some "similar" samples in the unlabeled data set, so that supervised techniques are made possible by such an enlarged labeled set.

3.2 Expectation-Discrimination-Maximization

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 calculated based on the probabilistic label $\mathbf{l}_j = \{l_{jk}, k = 1, \ldots, C\}$ assigned by this weak classifier.

$$l_{jk} = \frac{p(\mathbf{x}_j|c_k)p(c_k)}{\sum_{k=1}^{C} p(\mathbf{x}_j|c_k)p(c_k)} \tag{8}$$

$$w_{jk} = \lg(p(\mathbf{x}_j|c_k)p(c_k)) \ k = 1, \ldots, C \tag{9}$$

Every unlabeled sample will be weighted by its Mahalanobis distance to the class center. Euqation(9) is just a heuristic to weight unlabeled data $\mathbf{x}_j \in \mathcal{U}$, although there may be many other choices.

After that, MDA 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 linearly projected to a new space of dimension $C - 1$ but unchanging the labels and weights

$$\hat{\mathcal{D}} = \{\mathbf{W}^T\mathbf{x}_j, y_j : \forall \mathbf{x}_j \in \mathcal{L}\} \bigcup \{\mathbf{W}^T\mathbf{x}_j, \mathbf{l}_j, \mathbf{w}_j : \forall \mathbf{x}_j \in \mathcal{U}\}.$$

Then parameters Θ of the probabilistic models are estimated on $\hat{\mathcal{D}}$, so that the probabilistic labels are given by the Bayesian classifier according to Equation (8). The algorithm iterates over these three steps: expectation, discrimination,

and maximization. Figure 2 describes the D-EM algorithm. Generally, we use Gaussian or second-order Gaussian mixtures. Our experiments show that this algorithm works better than pure EM.

Discriminant-EM algorithm (D-EM)

inputs: labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$

output: classifier with parameters Θ

begin Initialize: number of components C

$\mathbf{W} \leftarrow MDA(\mathcal{L})$;

$lset \leftarrow \texttt{Projection}(\mathbf{W}, \mathcal{L})$;

$uset \leftarrow \texttt{Projection}(\mathbf{W}, \mathcal{U})$;

$\Theta \leftarrow \texttt{MAP}(lset)$;

D-E-M iteration {

E-step:

$plabel \leftarrow \texttt{Labeling}(\Theta, uset)$;

$weight \leftarrow \texttt{Weighting}(plabel)$;

$\mathcal{D}' \leftarrow \mathcal{L} \bigcup \{\mathcal{U}, plabel, weight\}$;

D-step:

$\mathbf{W} \leftarrow \texttt{MDA}(\mathcal{D}')$;

$lset \leftarrow \texttt{Projection}(\mathbf{W}, \mathcal{L})$;

$uset \leftarrow \texttt{Projection}(\mathbf{W}, \mathcal{U})$;

$\hat{\mathcal{D}} \leftarrow lset \bigcup \{uset, plabel, weight\}$;

M-step:

$\Theta \leftarrow \texttt{MAP}(\hat{\mathcal{D}})$;

}

return Θ;

end

Fig. 2. The D-EM algorithm.

It should be noted that the simplification of probabilistic structures in the mapped data space is not guaranteed by linear MDA. If the components of data distribution are mixed up, it is very unlikely to find such a linear mapping. In this case, nonlinear mapping should be found so that a simple probabilistic structure could be used to approximate the data distribution in the mapped data space.

4 The Kernel D-EM Algorithm

In this section, we try to extend the linear discriminant analysis to nonlinear analysis, in order to achieve better discrimination power. We take a kernel-based approach. The linear D-EM algorithm presented in the previous section will be generalized to the kernel D-EM algorithm in this section.

4.1 Nonlinear 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. In the F-space, a linear mapping $\mathbf{V}$ will be determined by MDA. Thus, we have

$$\mathbf{V}_{opt} = \arg\max_{\mathbf{V}} \frac{|\mathbf{V}^T S_B^{\phi} \mathbf{V}|}{|\mathbf{V}^T S_W^{\phi} \mathbf{V}|}, \tag{10}$$

where

$$S_B^{\phi} = \sum_{j=1}^{C} n_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T, \tag{11}$$

$$S_W^{\phi} = \sum_{j=1}^{C} \sum_{k=1}^{n_j} (\phi(\mathbf{x}_k) - \mathbf{m}_j)(\phi(\mathbf{x}_k) - \mathbf{m}_j)^T, \tag{12}$$

with $\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \phi(\mathbf{x}_k)$, $\mathbf{m}_j = \frac{1}{n_j} \sum_{k=1}^{n_j} \phi(\mathbf{x}_k)$, where $j = 1, \ldots, C$.

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 (Vapnik, 1995), principal components analysis (Schölkopf et al., 1998), and invariant feature extraction (Mika et al., 2000; Roth and Steinhage, 2000). 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 10, 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 $\underline{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$,

$$\mathbf{v} = \sum_{k=1}^{n} \alpha_k \phi(\mathbf{x}_k) = \Phi \underline{\alpha}, \tag{13}$$

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):

$$\mathbf{v}^T \phi(\mathbf{x}_k) = \underline{\alpha}^T \Phi^T \phi(\mathbf{x}_k) \tag{14}$$

$$= \underline{\alpha}^T \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix} = \underline{\alpha}^T \xi_k, \tag{15}$$

where

$$\xi_k = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix}, \tag{16}$$

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:

$$\mathbf{v}^T \mathbf{m}_j = \underline{\alpha}^T \frac{1}{n_j} \sum_{k=1}^{n_j} \begin{bmatrix} \phi^T(\mathbf{x}_1)\phi(\mathbf{x}_k) \\ \vdots \\ \phi^T(\mathbf{x}_n)\phi(\mathbf{x}_k) \end{bmatrix} \tag{17}$$

$$= \underline{\alpha}^T \begin{bmatrix} \frac{1}{n_j} \sum_{k=1}^{n_j} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ \frac{1}{n_j} \sum_{k=1}^{n_j} k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix} \tag{18}$$

$$= \underline{\alpha}^T \mu_j. \tag{19}$$

It follows that

$$\mathbf{v}^T S_B \mathbf{v} = \underline{\alpha}^T K_B \underline{\alpha}, \tag{20}$$

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

$$\mathbf{v}^T S_W \mathbf{v} = \underline{\alpha}^T K_W \underline{\alpha}, \tag{21}$$

where $K_W = \sum_{j=1}^{c} \sum_{k=1}^{n_j} (\xi_k - \mu_j)(\xi_k - \mu_j)^T$. The goal of Kernel Multiple Discriminant Analysis (KMDA), then, is to find

$$\mathbf{A}_{opt} = \arg\max_{\mathbf{A}} \frac{|\mathbf{A}^T K_B \mathbf{A}|}{|\mathbf{A}^T K_W \mathbf{A}|}, \tag{22}$$

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 the training set, the nonlinear mapping is dependent on the entire training samples. For large n, the solution to the generalized eigensystem is costly. A simple to approximate the 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.

Different from randomly picking a subset of training data, we maintain a set of kernel vectors at every iteration which are meant to be the key pieces of data for training. At the beginning, 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. Such an evolutionary kernel vector selection algorithm is summarized below in Figure 3.

4.3 The Kernel D-EM algorithm

We now apply KMDA to D-EM. *Kernel D-EM (KDEM)* is a generalization of D-EM, in which instead of using a simple linear transformation of the data, KMDA is used to project the data to a nonlinear subspace 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 $\mathcal{X}$ 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, i.e.,

$$p(l|\Theta) = \sum_{j=1}^{C} p(\mathbf{V}^T\phi(\mathbf{x})|c_j;\theta_j)p(c_j|\theta_j). \tag{23}$$

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 $KV = \{\nu_i, i = 1, \ldots, M\}$.

$k = 0$; $e = \infty$; $KV^{(0)}$ =`random_pick(`X`)`; // Init

`do`{

  $\mathbf{A}_{opt}^{(k)}$ =`KMDA(`$\underline{X}, KV^{(k)}$`)`;// Perform KMDA

  $Y^{(k)}$ =`Proj(`$X, \mathbf{A}_{opt}^{(k)}$`)`; // Project $\mathcal{X}$ to Δ

  $\Theta^{(k)}$ =`Bayes(`$Y^{(k)}, L$`)`; //Bayesian classifier

  $\bar{L}^{(k)}$ =`Labeling(`$Y^{(k)}, \Theta^{(k)}$`)`; // Classification

  $e^{(k)}$ =`Error(`$\bar{L}^{(k)}, L$`)`; // Calculate error

  `if(`$e^{(k)} < e$`)`

    $e = e^{(k)}$; $KV = KV^{(k)}$; $k = k + 1$;

    $KV^{(k)}$ =`random_pick(`$\{\mathbf{x}_i : \bar{l}_i^{(k)} \neq l_i\}$`)`;

  `else`

    $KV = KV^{(k-1)}$; `break`;

  `end`

}

`return` KV;

Fig. 3. Evolutionary kernel vector selection.

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. Then, the three steps of kernel

D-EM are iterated until some appropriate convergence criterion:

- E-step: set $\hat{\mathcal{Z}}^{(k+1)} = E[\mathcal{Z}|\mathcal{D}; \hat{\Theta}^{(k)}]$
- D-step: set $\mathbf{A}_{opt}^{k+1} = \arg\max_A \frac{|\mathbf{A}^T K_B \mathbf{A}|}{|\mathbf{A}^T K_W \mathbf{A}|}$, and identify kernel vectors $KV^{(k+1)}$
- 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. As mentioned before, this assumes that the class distributions are moderately smooth.

5 Experiments

In this section, we initially compare the KMDA algorithm with other supervised learning techniques on some standard data sets. Experimental results of the D-EM algorithm 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 [2] were used in our experiments. The benchmark data has 100 different realizations. In (Mika et al., 2000), 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), the AdaBoost algorithm, and the kernel Fisher discriminant

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

(KFD) (Mika et al., 1999). The RBF kernels were used in all kernel-based algorithms.

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	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

In Table 1, #-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 entire 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 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 labeled an image database of 134 images, which was 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 were 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 were obtained by relevance feedback. When using more than 100 unlabeled samples, the error rates dropped to less than 10%. From Figure 4, we found 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.

We tested and compared four methods. The first one was to weight each features by relevance feedback (WRF) (Rui et al., 1998), in which 37 image features, such as color moments, edge distribution features, and texture features, which were pre-calculated and pre-stored. The top 20 most similar images were obtained through ranking each image by comparing the Mahalanobis distances to the means of query images. The second method was a simple probabilistic method (SP), in which both classes (relevant and irrelevant) were assumed Gaussian distributions, and the model parameters were estimated by feedback images. The third method was the basic EM (EM) algorithm, which assumed Gaussian distributions for both classes. The fourth was the D-EM algorithm.

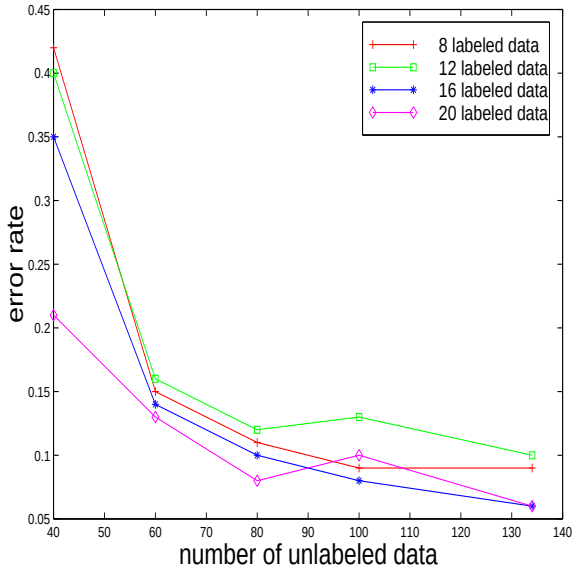


Fig. 4. The effect of labeled and unlabeled data in D-EM. Error rate decreases when adding more unlabeled data. Combining some unlabeled data can largely reduce the classification error.

In the last three probabilistic methods, the label of each image was given by maximizing *a posteriori* probability, $l_j = \arg\max_k p(c_k|\mathbf{x}_j)$.

We also compared the image features (I-Features) and the eigen features (E-Features). We used the same image features as that in WRF (Rui et al., 1998), in which 9 color features included the mean, std and skew of the HSV space, 10 texture features are extracted by wavelets, and 18 structure features were represented by the statistics of the edge map. The eigen features were extracted by PCA, in which the number of principal components is 30, and the resolution of image was reduced to 20×20. Except for WRF, both I-Features and E-Features were tested.

These four methods were compared on this fully labeled database. Classification error for each method was calculated for evaluation, although these errors were not available for the training. Suppose the database has N samples, C

classes, and the k-th class has N_k samples, and $N = \sum_{k=1}^{C} N_k$. The method to calculate error in WRF is different from the other three methods. In WRF, if the query images are drawn from a class, e.g., the j-th class, and m samples in the top N_j retried images belongs to the j-th class, the error for this query is defined as $e = 2(N_j - m)/N$. In the other three methods, if there are m samples in total that are not correctly labeled, the error is defined as $e = m/N$. The average error is obtained by averaging over M experiments.

Table 2

Error rate comparison among different algorithms. All comparisons are based on the first time relevance feedback with 6 relevant and 6 irrelevant images. D-EM outperforms the other three methods.

Algorithm	I-Features	E-Features
WRF	6.3%	N/A
SP	21.2%	15.7%
EM	23.4%	25.8%
D-EM	3.9%	5.3%

Our algorithm was also tested by several large databases. For example, 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. Since it is difficult to obtain quantitative classification results on these large databases, subjective evaluation is performed instead. Satisfactory results were obtained.

5.3 View-independent Hand Posture Recognition

Next, we examine results of the linear and kernel D-EM algorithms on a hand gesture recognition task. The task is to classify among 14 different hand postures, each of which represents a gesture commanding 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 videos of subjects making each of the hand postures. These 560 labeled images are used to test the classifiers by calculating the classification errors.

Hands were localized in video sequences by adaptive color segmentation and hand regions were cropped and converted to gray-level images (Wu and Huang, 2000). Gabor wavelet filters with 3 levels and 4 orientations were used to extract 12 texture features. 10 coefficients from the Fourier descriptor of the occluding contour were used to represent hand shape. We also used 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 represented 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) (Wu and Huang, 2000). In our experiments, we used 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 3 shows the comparison.

We observed that multilayer perceptrons were often trapped in local minima and nearest neighbor suffers from the sparsity of the labeled templates. The

Table 3

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

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%

poor performance of the standard EM was due to the fact that the generative model did not capture the ground-truth distribution well, since the underlying data distribution was highly complex. It is not surprising that the linear D-EM and the kernel D-EM algorithm outperformed other methods, since the D-step optimizes separability of the classes. Finally, note the effectiveness of the kernel D-EM. We find that KDEM often appears to project classes to approximately Gaussian clusters in the transformed space, which facilitates their modeling with Gaussians.

5.4 Nonstationary Color Tracking

This section presents our experiments of the Discriminant-EM algorithm for nonstationary color tracking. A Gaussian mixture model was used to model color distributions for color segmentation. The D-EM algorithm was employed to transduce a prior color model to new images for better segmentation then better tracking results.

(a)

(b) (c)

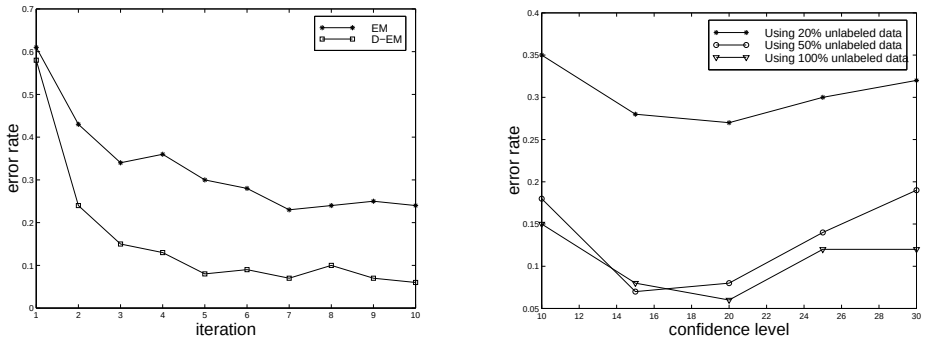
Fig. 5. (a) Some correctly classified images by both LDEM and KDEM (b) images that are mislabeled by LDEM, but correctly labeled by KDEM (c) images that neither LDEM or KDEM can correctly labeled.

5.4.1 Simulation

At current time t in tracking, since M_{t-1} may not be able to give a good segmentation on I_t, the image at time t is not labeled (segmented) so that the ground truth for the new data set is not available. However, to evaluate our algorithm, we assume the ground truth is known to calculate classification errors, although such errors are not available in real applications.

We used two "hand images" (resolution 100×75), where I_1 was a segmented image, and I_2 was the same as I_1 except that the color distribution of I_2 was transformed by shifting the R element of every pixel by 20 such that I_2 looks like adding a red filter. A color classifier was learned for I_1 with error rate less than 5%. In this simple situation, this color classifier would fail to correctly

segment hand region from I_2, since the skin color in I_2 is much different. Actually, it had error rate of 35.2% on I_2.



(a) (b)

Fig. 6. (a) shows the comparison between EM and D-EM. (b) shows the effect of number of labeled and unlabeled data in D-EM

Figure 6.a shows the comparison between EM and D-EM. In this experiment, both EM and D-EM converged after several iterations, but D-EM gave a lower classification error rate (6.9% vs. 24.5%). To investigate the effect of the unlabeled data used in D-EM, we fed the algorithm a different number of labeled and unlabeled samples. The number of labeled data is controlled by the confidence level. In this experiment, confidence level was the same as the size of the labeled set. In general, combining unlabeled data can largely reduce the classification error when labeled data are very few. When using 20% (1500) unlabeled data, the lowest error rate achieved was 27.3%. When using 50% (3750) unlabeled data, the lowest error rate dropped to 6.9%. The transduced color classifier gave around 30% more accuracy. Figure 6.b shows the effect of different sizes of labeled and unlabeled data sets in D-EM.

5.4.2 Hand and Face Localization

This color tracking algorithm is applied to a gesture interface, in which hand gesture commands are localized and recognized to serve as inputs of a virtual environment application. These experiments ran at 15-20Hz on a single processor SGI O2 R10000 workstation.

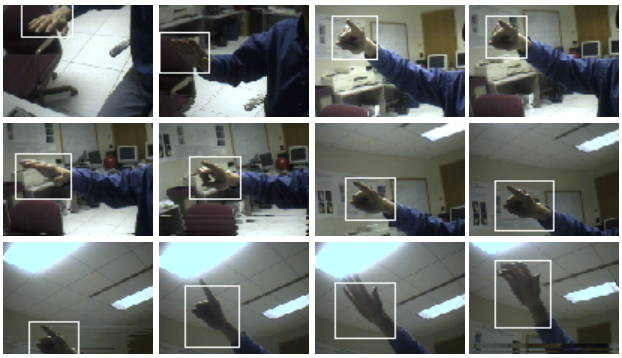


Fig. 7. Hand Localization by D-EM

Fig. 8. Face localization by D-EM

Figure 7 and Figure 8 show two examples of hand and face localization in a typical lab environment. Both cases are difficult for static color models. In Figure 7, the skin color in different parts of hand are different. The camera moves from downwards to upwards and the lighting conditions on the hand are different. Hand becomes darker when it shades the light sources in several frames. In Figure 8, skin color changes a lot when the head moves back and forth, and turns around.

6 Conclusion and Future Work

Many visual learning tasks are confronted by some common difficulties, such as the lack of large supervised training data sets and the learning in high dimensional spaces. In this paper, we presented a self-supervised learning technique, the Discriminant-EM algorithm, 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 showed that KMDA with these adaptations performs comparably with the best known supervised learning algorithms. On our real experiments for recognizing hand postures, content-based image retrieval and nonstationary color tracking, D-EM outperformed some naïve supervised learning methods and existing semi-supervised algorithms.

In this paper, we did not show the convergence properties of the D-EM algorithm. Indeed, we did observe divergence cases when the labeled examples were badly picked. However, we did not figure out the exact reason behind the divergence, i.e., the conditions that guarantees the convergence of the D-EM iterations. The relationship between the labeled and unlabeled sets will effect the convergence, which is yet to be explored. We are going to investigate these issues in our further study. In addition, examination of the experimental results reveals that KMDA often maps data sets corresponding to each class into approximately Gaussian clusters in the transformed 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 Northwestern Faculty Startup Funds, National Science Foundation Grants CDA-96-24396 and EIA-99-75019 and NSF Alliance Program. The authors also would like to thank the anonymous reviewers for their constructive comments.

Ying Wu received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1994, the M.S. degree from Tsinghua University, Beijing, China, in 1997, and the Ph.D. degree in electrical and computer engineeringfrom the University of Illinois at Urbana-Champaign (UIUC), Urbana, Illinois, in 2001. From 1997 to 2001, he was a Graduate Research Assistant at the Image Formation and Processing Group of the Beckman Institute for Advanced Science and Technology at UIUC. During summer 1999 and 2000, he was a Research Intern with the Vision Technology Group, Microsoft Research, Redmond, Washington. Since 2001, he had been on the faculty of the Department of Electrical and Computer Engineering at the Northwestern University, Evanston, Illinois. His current research interests include computer vision, computer graphics, machine learning, human-computer intelligent interaction, image/video processing, multimedia, and virtual environments. He received the Robert T. Chien Award at the University of Illinois at Urbana-Champaign in 2001.

Thomas S. Huang received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, China, and

the M.S. and Sc.D. degrees in electrical engineering from Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts. He was on the Faculty of Department of Electrical Engineering at MIT from 1963 to 1973 and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now William L. Everitt Distinguished Professor of Electrical and Computer Engineering, Research Professor at the Coordinated Science Laboratory, and Head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology. During his sabbatical leaves, he has worked at the MIT Lincoln Laboratory, the IBM Thomas J. Watson Research Center, and Rheinishes Landes Museum in Bonn, West Germany, and he held Visiting Professor positions at the Swiss Institutes of Technology in Zurich and Lausanne, the University of Hannover, Germany, INRS-Telecommunications of the University of Quebec, Montreal, Canada, and the University of Tokyo, Japan. He has served as a consultant to numerous industrial firms and government agencies both in the United States and abroad. His professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 12 books and over 400 papers in network theory, digital filtering, image processing, and computer vision. He is a Founding Editor of the *International Journal Computer Vision, Graphics and Image Processing* and Editor of the "Springer Series in Information Sciences," published by Springer Verlag. Dr. Huang is a Fellow of the International Association of Pattern Recognition, and the Optical Society of American and has received a Guggenheim Fellowship, an A. V. Humboldt Foundation Senior U.S. Scientist Award, and a Fellowship from the Japan Association for the Promotion of Science. He received the IEEE Acoustics, Speech and Signal

Processing Society's Technical Achievement Award in 1987 and the Society Award in 1991. He received the IEEE Signal Processing Society's Technical Achievement Award in 1987, and the Society Award in 1991. He was awarded the IEEE Third Millennium Medal in 2000. He received the Honda Lifetime Achievement Award for "contributions to motion analysis" in 2000. He received the IEEE Jack S. Kilby Medal in 2001. Dr. Huang is a member of the National Academy of Engineering, and a Foreign Member of the Chinese Academy of Engineering.

References

Basri, R., Roth, D., Jacobs, D., June 1998. Clustering appearances of 3D objects. In: Proc. of IEEE Conf. Computer Vision and Pattern Recognition. Santa Barbara, CA, pp. 414–420.

Belhumeur, P., Hespanha, J., Kriegman, D., April 1996. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In: Proc. of European Conference on Computer Vision. Cambridge, UK.

Bennett, K., 1999. Combining support vector and mathematical programming methods for classification. In: Schoelkopf, A., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods - Support Vector Learning. Cambridge, MA: MIT Press, pp. 307–326.

Blum, A., Mitchell, T., July 1998. Combining labeled and unlabeled data with co-training. In: Proc. Conf. Computational Learning Theory. Madison, WI, pp. 92–100.

Cui, Y., Weng, J., 1996. Hand segmentation using learning-based prediction and verification for hand sign recognition. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. San Francisco, CA, pp. 88–93.

de Sa, V. R., 1994. Learning classification with unlabeled data. In: Cowan, J.,

Tesauro, G., Alspector, J. (Eds.), Advances in Neural Infomation Processing Systems 6. Morgan-Kaufmann, pp. 112–119.

de Sa, V. R., Ballard, 1998. Category learning through multi-modality sensing. Neural Computation 10 (5).

Duda, R., Hart, P., 1973. Pattern Classification and Scene Analysis. Wiley, New York.

Gammerman, A., Vapnik, V., Vowk, V., July 1998. Learning by transduction. In: Proc. of Conf. Uncertainty in Artificial Intelligence. Madison, WI, pp. 148–156.

Ghahramani, Z., Jordan, M. I., 1994. Supervised learning from incomplete data via án EM approach. In: Cowan, J., Tesauro, G., Alspector, J. (Eds.), Advances in Neural Information Processing Systems 6. Morgan-Kaufmann, pp. 120–127.

Joachims, T., June 1999. Transductive inference for text classification using support vector machines. In: Proc. of Int'l Conf. on Machine Learning. Bled, Slovenia, pp. 200–209.

Jones, M., Rehg, J., 1998. Statistical color models with application to skin detection. Tech. Rep. CRL-98-11, Compaq Cambridge Research Lab., Cambridge, MA.

Manjunath, B., Ma, W. Y., Aug. 1996. Texture features for browsing and retrieval of image data. IEEE Trans. on Pattern Ananlysis and Machine Intelligence 18, 837–842.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., Müller, K.-R., August 1999. Fisher discriminant analysis with kernels. In: IEEE workshop on Neural Networks for Signal Proceesing. Madison, WI.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., Müller, K.-R., 2000. Invariant feature extraction and classification in kernel spaces. In: Solla, S., Leen, T., Muller, K. (Eds.), Advances in Neural Information Processing Systems 12. MIT Press, Cambridge, MA, pp. 526–532.

Mitchell, T., 1999. The role of unlabeled data in supervised learning. In: Proc. Sixth Int'l Colloquium on Cognitive Science. Spain.

Nigam, K., McCallum, A., Thrun, S., Mitchell, T., 1999. Text classification from labeled and unlabeled documents using EM. Machine Learning 39 (2/3), 103–134.

Popescu, M., Gader, P., 1998. Image content retrieval from image database using feature integration by choquet integral. In: Proc. SPIE Storage and Retrieval for Image and Video Database. Vol. VII. San Jose, CA.

Raja, Y., McKenna, S., Gong, S., June 1998. Colour model selection and adaptation in dynamic scenes. In: Proc. of European Conf. on Computer Vision. Freiburg, Germany, pp. 460–475.

Riloff, E., Jones, R., July 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In: Proc. AAAI National Conf. on Artificial Intelligence. Orlando, FL, pp. 1044–1049.

Roth, V., Steinhage, V., 2000. Nonlinear discriminant analysis using kernel functions. In: Solla, S., Leen, T., Muller, K. (Eds.), Advances in Neural Information Processing Systems. MIT Press, Cambridge, MA, pp. 568–574.

Rui, Y., Huang, T. S., Ortega, M., Mehrotra, S., 1998. Relevance feedback: A power tool for interactive content-based image retrieval. IEEE Trans. on Circuits and Systems for Video Technology 8, 644–655.

Santini, S., Jain, R., 1999. Similarity measures. IEEE Trans. on Pattern Analysis and Machine Intelligence 21, 871–883.

Schölkopf, B., Smola, A., Robert Müller, K., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319.

Swets, D., Weng, J., 1999. Hierarchical discriminant analysis for image retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence 21, 386–400.

Tresp, V., Neuneier, R., Ahmad, S., 1995. Efficient methods for dealing with missing data in supervised learning. In: Tesauro, G., Touretzky, D., Leen, T. (Eds.), Advances in Neural Information Processing Systems 7. MIT Press, Cambridge, MA, pp. 689–695.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New

York.

Weber, M., Welling, M., Perona, P., 2000. Towards automatic discovery of object categories. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Vol. 2. Hilton Head Island, South Carolina, pp. 101–108.

Wu, Y., Huang, T. S., June 2000. View–independent recognition of hand postures. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Vol. II. Hilton Head Island, South Carolina, pp. 88–94.

Wu, Y., Huang, T. S., July 2001. Robust visual tracking by co-inference learning. In: Proc. IEEE Int'l Conference on Computer Vision. Vol. II. Vancouver, pp. 26–33.

Wu, Y., Tian, Q., Huang, T. S., June 2000. Discriminant-EM algorithm with application to image retrieval. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Vol. I. Hilton Head Island, South Carolina, pp. 222–227.

Wu, Y., Toyama, K., Huang, T. S., July 2001. Self-supervised learning for object recognition based on kernel Discriminant-EM algorithm. In: Proc. IEEE Int'l Conference on Computer Vision. Vol. I. Vancouver, pp. 275–280.