Self-Supervised Learning for Visual Tracking and Recognition of Human Hand

Ying Wu, Thomas S. Huang

Beckman Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{yingwu, huang}@ifp.uiuc.edu

Abstract

Due to the large variation and richness of visual inputs, statistical learning gets more and more concerned in the practice of visual processing such as visual tracking and recognition. Statistical models can be trained from a large set of training data. However, in many cases, since it is not trivial to obtain a large labeled and representative training data set, it would be difficult to obtain a satisfactory generalization. Another difficulty is how to automatically select good features for representation. By combining both labeled and unlabeled training data, this paper proposes a new learning paradigm, self-supervised learning, to investigate the issues of learning bootstrapping and model transduction. Inductive learning and transductive learning are the two main cases of self-supervised learning, in which the proposed algorithm, Discriminant-EM (D-EM), is a specific learning technique. Vision-based gesture interface is employed as a testbed in our research.

Introduction

In current Virtual Environment (VE) applications, some conventional interface devices, such as keyboards, mice, wands and joysticks, are inconvenient and unnatural. In recent years, the use of hand gestures in human computer interaction serves as a motivating force for research in hand tracking and gesture recognition. Although hand gestures are complicated to model since the meanings of hand gestures depend on people and cultures, a set of specific hand gesture vocabulary can be always predefined in many applications, so that the ambiguity can be limited. Hand tracking and posture recognition are two of the main components in vision-based gesture interface.

One goal of hand tracking is to locate hand regions in video sequences. Skin color offers an effective and efficient way to segment hand regions out. According to the representation of color distribution in certain color spaces, current techniques of color tracking can be classified into two general approaches: non-parametric (Swain and Ballard 1991; Kjeldsen and Kender 1996; Jones and Rehg 1998; Wu, Liu, and Huang 2000) and parametric (Raja, McKenna, and Gong 1998). Many different color spaces, such as RGB, HSV, N-RGB, have been used in current research. However, many of these techniques are plagued by some special difficulties such as large variation in skin tone, unknown lighting conditions and dynamic scenes.

One possible solution is to make a generic statistical skin color model by collecting a huge training data set (Jones and Rehg 1998) so that the generic color model could work for any user in any case. However, collecting and labeling such a huge database is not trivial. Even though such a good generic color model can be obtained, the skin color may looks very different in different lighting conditions. This color constancy problem is not trivial in color tracking. Because of dynamic scenes and changing lighting conditions, the color distribution over time is non-stationary, since the statistics of color distribution will change with time. If a color classifier is trained under a specific condition, it may not work well in other scenarios.

In many gesture interfaces, some simple controlling, commanding and manipulative gestures are defined to fulfill natural interaction such as pointing, navigating, moving, rotating, stopping, starting, selecting, etc. View-independent hand posture recognition is to recognize hand signs even from different viewing directions.

One approach is the 3D model-based approach, in which the hand configuration is estimated by taking advantage of 3D hand models (Davis and Shah 1994; Heap and Hogg 1996; Kuch and Huang 1995; Lee and Kunii 1995; Rehg and Kanade 1995; Wu and Huang 1999). Since hand configurations are independent to view directions, these methods could directly achieve view-independent recognition. However, since a classification of hand postures is often enough in many other applications such as commands switching, an alternative approach is appearance-based approach (Cui and Weng 1996; Quek and Zhao 1996; Triesch and von de Malsburg 1996), in which classifiers are learned from a set of image samples. Although it is easier for the appearance-based approach to achieve user-independence than model-based approach, there are two major difficulties of this approach: automatic feature selection and training data collection. In general, good generalization requires a large and representative labeled training data set. However, to manually label a large data set will be very time-consuming and tedious. Although unsupervised schemes has been proposed to clustering the appearances of 3D objects(Basri, Roth, and Jacobs 1998), it is hard for pure unsupervised approach to achieve accurate classification without supervision.

In this paper, color tracking is formulated as a transduc-

tive learning problem, and posture recognition is formulated as an inductive learning problem. These two learning problems are unified in a framework of self-supervised learning in which both supervised and unsupervised training data are employed.

Problem Formulation

Unlabeled Data

Traditionally, feature extraction and selection are independent to the designation of classifier. Although the discriminant analysis technique offers a means to automatically select and weight classification-relevant features, it puts a harsh requirement to the training data set: a large labeled data set. We do not expect discriminant analysis to output a good result, unless enough labeled data are available.

In fact, it seems that it might not be necessary to have every sample labeled in supervised learning. A very interesting result given by the theory of the support vector machine (SVM) (Vapnik 1995) is that the classification boundary is related only to some support vectors, rather than the whole data set. Although the identification of these support vectors is not trivial, it motivates us to think about the roles of non-support vectors. Fortunately, it is easier to collect unlabeled data. 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 (Gammerman, Vapnik, and Vowk 1998; Bennett and Demiriz 1998; Joachims 1999). However, when the size of unlabeled data becomes very large, these methods need formidable computational resources for mathematical programming. Another difficulty of these SVM-based methods is that the way of selecting the kernel function is heuristic. Some other alternative methods try to fit this problem into the EM framework and employ parametric models (Mitchell 1999; Nigam *et al.* 1999), and have some applications in text classification. 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.

If the probabilistic structure of data distribution is known, parameters of probabilistic models can be estimated by unsupervised learning alone, but it is still impossible to assign class labels without labeled data (Duda and Hart 1973). This fact suggests that labeled and unlabeled training data are both needed in learning, in which labeled data (if enough) can be used to label the class and unlabeled data can be used to estimate the parameters of generative models.

Self-supervised Learning

In self-supervised learning, there is a hybrid training data set $\mathcal{D}$ which consists of a labeled data set $\mathcal{L} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$, where $\mathbf{x}_i$ is feature vector, y_i is label and N is the size of the set, and an unlabeled data set $\mathcal{U} = \{\mathbf{x}_i, i = 1, \ldots, M\}$, where M is the size of the set. Generally, we make an assumption here that $\mathcal{L}$ and $\mathcal{U}$ are from the same distribution. Essentially, the classification problem can be represented as:

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

where Ψ is a subset of the whole data space Ω and C is the number of classes. According to different Ψ, self-supervised learning has different special cases.

The Inductive Problem When $\Psi = \Omega$, self-supervised learning becomes inductive learning. The classification is represented as:

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

Different from conventional learning paradigms, inductive learning depends on both supervised data set $\mathcal{L}$ and unsupervised data set $\mathcal{U}$. If $\mathcal{L} = \phi$, it degenerates to pure unsupervised learning. If $\mathcal{U} = \phi$, it degenerates to pure supervised learning. Generally, we use a large unlabeled training set and a relatively small labeled set.

The Transductive Problem When $\Psi = \mathcal{U}$, self-supervised learning becomes transductive learning. The classification is represented as:

$$y_i = arg \max_{j=1,\ldots,C} p(y_j|\mathbf{x}_i, \mathcal{L}, \mathcal{U} : \forall \mathbf{x}_i \in \mathcal{U}) \qquad (3)$$

Generally, the classifier obtained from inductive learning could be highly nonlinear, and a huge labeled training set is required to achieve good generalization. However, the requirement of generalization could be relaxed to a subset of the whole data space. The generalization of transductive learning is only defined on the unlabeled training set $\mathcal{U}$, instead of the whole data space Ω.

It can be illustrated by an example of non-stationary color tracking, in which each color pixel will be labeled by a color classifier or model (M). In transductive learning, 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. 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:

$$y_i = arg \max_{j=1,\ldots,C} p(y_j|\mathbf{x}_i, M_t, I_{t+1} : \forall \mathbf{x}_i \in I_{t+1}) \qquad (4)$$

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 distribution 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. 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:

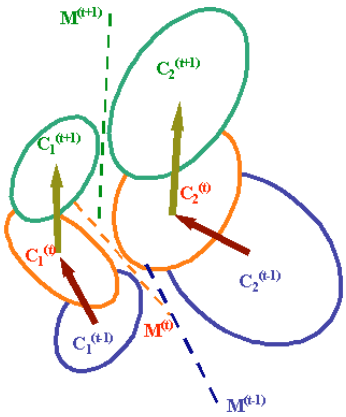


Figure 1: An illustration of transduction of classifiers.

labeled data set $\mathcal{L} = \{(\mathbf{x}_j, y_j), j = 1, \ldots, N\}$, and unlabeled set $\mathcal{U} = \{\mathbf{x}_j, j = 1, \ldots, M\}$. Here, $\mathcal{L}$ and $\mathcal{U}$ are from the same distribution. Consequently, the transductive classification can be written as 3. 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}.

Generative Model

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:

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{j=1}^{C} p(\mathbf{x}|c_j; \theta_j) p(c_j|\theta_j) \tag{5}$$

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. $\{y_j = c_j, j = 1, \ldots, C\}$.

The D-EM Algorithm

In this section, we describe the EM framework and the proposed D-EM algorithm to the self-supervised learning problem.

The EM Framework

Since the labels of unlabeled data can be treated as missing values, the Expectation-Maximization (EM) approach can be applied to this transductive learning problem. The training data set $\mathcal{D}$ is a union of a set of labeled data set $\mathcal{L}$ and a set of unlabeled set $\mathcal{U}$. When we assume sample independency, the model 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 $l(\mathbf{\Theta}|\mathcal{D}) = \lg(p(\mathbf{\Theta})p(\mathcal{D}|\mathbf{\Theta}))$, and we have

$$l(\mathbf{\Theta}|\mathcal{D}) = \lg(p(\mathbf{\Theta})) + \sum_{\mathbf{x}_i \in \mathcal{U}} \lg(\sum_{j=1}^{C} p(O_j|\mathbf{\Theta}) p(\mathbf{x}_i|O_j; \mathbf{\Theta}))$$

$$+ \sum_{\mathbf{x}_i \in \mathcal{L}} \lg(p(y_i = O_i|\mathbf{\Theta}) p(\mathbf{x}_i|y_i = O_i; \mathbf{\Theta})) \tag{6}$$

When introducing a binary indicator $\mathbf{z}_i = (z_{i1}, \ldots, z_{iC})$, where $z_{ij} = 1$ iff $y_i = O_j$, and $z_{ij} = 0$ otherwise, we have:

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

The EM algorithm estimates the parameters $\mathbf{\Theta}$ by an iterative hill climbing procedure, which alternatively calculates $E(\mathcal{Z})$, the expected values for 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 $\hat{\mathcal{Z}}^{(k+1)} = E[\mathcal{Z}|\mathcal{D}; \hat{\Theta}^{(k)}]$
- 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 $\mathbf{\Theta}$ at the k-th iteration respectively.

If the probabilistic structure, such as the number of components in mixture models, is known, EM could estimate true probabilistic model parameters. Otherwise, the performance could be very bad. A Gaussian distribution is often assumed to represent a class. Unfortunately, this assumption is often invalid in practice.

The D-EM Algorithm

Since we generally do not know the probabilistic structure of data distribution, EM often fails when structure assumption does not hold. One approach to this problem is to try every possible structure and select the best one. However, it needs 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 mixtures. The Multiple Discriminant Analysis (MDA) technique offers a way to relax the assumption of probabilistic structure, and EM supplies MDA a large labeled data set to select most discriminating features.

MDA is a natural generalization of Fisher's linear discrimination (LDA) in the case of multiple classes(Duda and Hart 1973). 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 of the between-class scatter and the within-class scatter is maximized in some sense. Details can be found in (Duda and Hart 1973). MDA offers a means to catch 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 enough labeled samples to estimate some

statistics such as mean and covariance. By combining MDA with the EM framework, our proposed method, Discriminant-EM algorithm (D-EM), is such a way to combine 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 in such an enlarged labeled set.

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.

$$l_{jk} = \frac{p(\mathbf{W}^T\mathbf{x}_j|c_k)p(c_k)}{\sum_{k=1}^{C} p(\mathbf{W}^T\mathbf{x}_j|c_k)p(c_k)} \tag{7}$$

$$w_{jk} = \lg(p(\mathbf{W}^T\mathbf{x}_j|c_k))\ k = 1, \ldots, C \tag{8}$$

Euqation(8) 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(7). The algorithm iterates over these three steps, "Expectation-Discrimination-Maximization". The following is the description of the D-EM algorithm.

Discriminant-EM algorithm (D-EM)
inputs: labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$
output: classifier with parameters $\mathbf{\Theta}$
begin Initialize: number of components C
  $\mathbf{W} \leftarrow MDA(\mathcal{L})$
  $lset \leftarrow Projection(\mathbf{W}, \mathcal{L})$
  $uset \leftarrow Projection(\mathbf{W}, \mathcal{U})$
  $\mathbf{\Theta} \leftarrow MAP(lset)$
  D-E-M iteration
    E-step:
      $plabel \leftarrow Labeling(\mathbf{\Theta}, uset)$
      $weight \leftarrow Weighting(plabel)$
      $\mathcal{D}' \leftarrow \mathcal{L} \bigcup \{\mathcal{U}, plabel, weight\}$
    D-step:
      $\mathbf{W} \leftarrow MDA(\mathcal{D}')$
      $lset \leftarrow Projection(\mathbf{W}, \mathcal{L})$
      $uset \leftarrow Projection(\mathbf{W}, \mathcal{U})$
      $\hat{\mathcal{D}} \leftarrow lset \bigcup \{uset, plabel, weight\}$
    M-step:
      $\mathbf{\Theta} \leftarrow MAP(\hat{\mathcal{D}})$
  return $\mathbf{\Theta}$
end

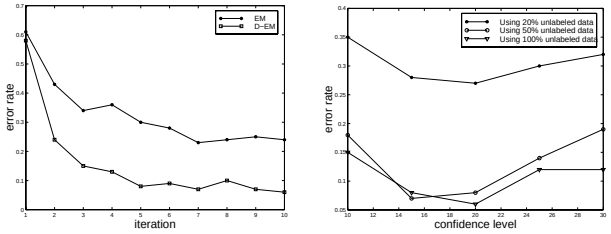
It should be noted that the simplification of probabilistic structures is not guaranteed in MDA. If the components of data distribution are mixed up, it is very unlikely to find such a linear mapping. In this case, nonlinear mapping should be found so that simple probabilistic structure could be used to approximate the data distribution in the mapped data space. Generally, we use Gaussian or 2-order Gaussian mixtures. Our experiments show that D-EM works better than pure EM.

Experiments

In our experiments, color tracking is formulated as a transductive problem that is described before, and hand posture recognition is treated as an inductive problem. The investigation of the effect of self-supervision and the effectiveness of D-EM are reported.

Color Tracking

Although these compact 3-D color spaces have substantial physical meanings, none of them is found to be able to give satisfactory color invariants through different lighting conditions. Considering that HSV color space is not a linear transformation of RGB space, we try to use a higher dimensional color space (6-D) by combining HSV and RGB spaces. In one of the experiments, to evaluate our algorithm in color tracking, we assume the segmentation is known to calculate classification errors, although such errors are not available in real applications. We use two "hand images" (resolution 100×75), where I_1 is a segmented image, and I_2 is the same as I_1 except that the color distribution of I_2 is transformed by shifting the R element of every pixel by 20 such that I_2 looks like adding a red filter. A color classifier is 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 has error rate of 35.2% on I_2.



(a) (b)

Figure 2: (a) shows the comparison between EM and D-EM. (b) shows the effect of number of labeled and unlabeled data in D-EM

Figure 2(a) shows the comparison between EM and D-EM, in which D-EM gives a lower classification error rate (6.9% vs. 24.5%). We feed 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 is 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 is 27.3%. When using 50% (3750) unlabeled data, the lowest error rate drops to 6.9%. The transduced color

classifier gives around 20% more accuracy, which is shown in Figure 2(b).

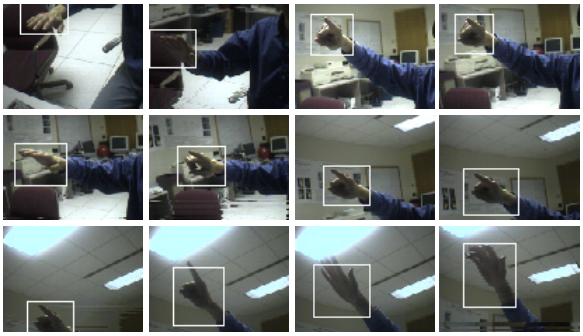


Figure 3: Hand Localization by D-EM

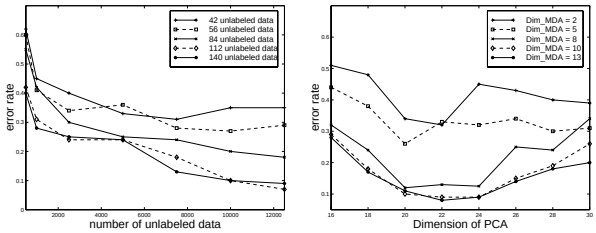
We also perform real experiments by implementing this tracking algorithm, which runs at 15-20Hz on a single processor SGI O2 R10000 workstation. Figure 3 shows an example of hand localization in a typical lab environment. In Figure 3, the skin color in different parts of hand are different. The camera 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.

Hand Posture Recognition

The gesture vocabulary in our gesture interface is 14. The hand localization system is employed to automatically collect hand images which serve as the unlabeled data, since the localization system only outputs bounding boxes of hand regions, regardless of hand postures. A large unlabeled database can be easily constructed. Currently, there are 14,000 unlabeled hand images in our database. It should be noted that the bounding boxes of some images are not tight, which introduce noise to the training data set. For each posture class, some samples are manually labeled. To investigate the effect of using unlabeled data and to compare different classification algorithms, we construct a testing data set, which consists of 560 labeled images.

Physical (P-) and mathematical (M-) features are both used as hand representation in our experiments. Gabor wavelet filters with 3 levels and 4 orientations are used to extract 12 texture features, each of which is the standard deviation of the wavelet coefficients from one filter. 10 coefficients from the Fourier descriptor are used to represent hand shapes. We also use some statistics such as the hand area, contour length, total edge length, density, and 2-order moments of edge distribution. Therefore, we have 28 low-level image features in total. After resizing the images to 20×20, some mathematical features are extracted by PCA.

We feed the algorithm a different number of labeled and unlabeled samples. In this experiment, we use 500, 1000, 2500, 5000, 7500, 10000, 12500 unlabeled samples and 42, 56, 84, 112, 140 labeled data, respectively. In this experiment, we use the mathematic features extracted by PCA with 22 principal components, and the dimension for MDA is set to 10. As shown in Figure 4(a), in general, combining some unlabeled data reduce the classification error by 20% to 30%.



(a) (b)

Figure 4: (a) shows the effect of labeled and unlabeled data in D-EM. (b) shows the effect of the dimension of PCA and MDA in D-EM

In Figure 4(b), we study the effect of the dimension parameters in PCA and MDA. If less principal components of PCA are used, some minor but important discriminating features may be neglected so that those principal components may be insufficient to discriminate different classes. On the other hand, if more principal components of PCA are used, it would include more noise. Therefore, the number of principal components of PCA is an important parameter for PCA. The dimension of MDA ranges between 1 to $C-1$, where C is the number of classes. We are interested in a lower dimensional space in which different classes can be classified. In this experiment, we use 112 labeled data and 10000 unlabeled data, and we find that a good dimension parameter of PCA is around 20 to 24, and 8 to 13 for MDA.

Four classification algorithms are compared in this experiment. For M-Features, the number of principal components of PCA is set to 22, and a set of 560 labeled data is used to perform MDA with dimension of 10. Using 1000 labeled training data, the multi-layer perceptron used in this experiment has one hidden layer of 25 nodes. We experiment with two schemes of the nearest neighbor classifier. One is just of 140 labeled samples, and the other uses 140 labeled samples to bootstrap the classifier by a growing scheme, in which newly labeled samples will be added to the classifier according to their labels. The labeled and unlabeled data for both EM and D-EM are 140 and 10000, respectively. Table 1 shows the comparison.

Algorithm	P-Features	M-Features
Multi-layer Perceptron	33.3%	39.6%
Nearest Neighbor	30.2%	35.7%
Nearest Neighbor(growing)	15.8%	20.3%
EM	21.4%	20.8%
D-EM	9.2%	7.6%

Table 1: Comparison among different algorithms

As shown in Table 1, the D-EM algorithm outperforms the other three methods. The multi-layer perceptron is often trapped in local minima in this experiment. The poor performance of the nearest neighbor classifier is partly due

to the insufficient labeled data. When the growing scheme is used, it reduces the error by 15%, since it automatically expends the stored templates. The problem of this scheme is that it is affected by the order of inputs, because there is no confidence measurement in growing so that the error of labeling will be accumulated. Pure EM algorithm hardly converges to a satisfactory classification in our experiments. However, D-EM ends up with a pretty good result.

Conclusion

This paper presents a study of a new learning paradigm, named self-supervised learning, which employs both supervised and unsupervised training data sets. Inductive learning and transductive learning can be treated as two special cases of this new learning paradigm. One possible approach in self-supervised learning is based on the EM framework. Integrating discriminant analysis and the EM framework, the proposed Discriminant-EM (D-EM) algorithm offers a means to relax the assumption of probabilistic structures of data distribution and automatically select a good classification features. In vision-based gesture interface, hand tracking and hand posture recognition offer two applications of self-supervised learning. Experiments show that the proposed D-EM algorithm outperforms some other learning techniques, and self-supervised has many potential applications.

One of the future research directions of this approach is to explore the nonlinear case of MDA. Like nonlinear SVM, some kernel functions should be studied. The convergence and stability analysis should be performed in our future research. Model transduction by using both labeled and unlabeled data is an interesting research topic, which needs more investigation.

Acknowledgments

This work was supported in part by National Science Foundation Grant IRI-9634618 and Grant CDA-9624396. The authors would like to appreciate the anonymous reviewers for their comments.

References

Basri, R.; Roth, D.; and Jacobs, D. 1998. Clustering appearances of 3D objects. In *Proc. of IEEE Int'l Conf. Computer Vision and Pattern Recognition*.

Bennett, K., and Demiriz, A. 1998. Semi-supervised support vector machines. In *Proc. of Neural Information Processing Systems*.

Cui, Y., and Weng, J. 1996. Hand sign recognition from intensity image sequences with complex background. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 88–93.

Davis, J., and Shah, M. 1994. Visual gesture recognition. *Vision, Image, and Signal Processing* 141:101–106.

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

Gammerman, A.; Vapnik, V.; and Vowk, V. 1998. Learning by transduction. In *Proc. of Conf. Uncertainty in Artificial Intelligence*, 148–156.

Heap, T., and Hogg, D. 1996. Towards 3D hand tracking using a deformable model. In *Proc. of IEEE Int'l Conf. Automatic Face and Gesture Recognition*.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proc. of Int'l Conf. on Machine Learning*.

Jones, M., and Rehg, J. 1998. Statistical color models with application to skin detection. Technical Report CRL 98/11, Compaq Cambridge Research Lab.

Kjeldsen, R., and Kender, J. 1996. Finding skin in color images. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 312–317.

Kuch, J. J., and Huang, T. S. 1995. Vision-based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *Proc. of IEEE Int'l Conf. on Computer Vision*, 666–671.

Lee, J., and Kunii, T. 1995. Model-based analysis of hand posture. *IEEE Computer Graphics and Applications* 77–86.

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

Nigam, K.; McCallum, A.; Thrun, S.; and Mitchell, T. 1999. Text classification from labeled and unlabeled documents using EM. *Machine Learning*.

Quek, F., and Zhao, M. 1996. Inductive learning in hand pose recognition. In *Proc. of IEEE Int'l Conf. on Automatic Face and Gesture Recognition*.

Raja, Y.; McKenna, S.; and Gong, S. 1998. Colour model selection and adaptation in dynamic scenes. In *Proc. of European Conf. on Computer Vision*.

Rehg, J., and Kanade, T. 1995. Model-based tracking of self-occluding articulated objects. In *Proc. of IEEE Int'l Conf. Computer Vision*, 612–617.

Swain, M., and Ballard, D. 1991. Color indexing. *Int. J. Computer Vision* 7:11–32.

Triesch, J., and von de Malsburg, C. 1996. Robust classification of hand postures against complex background. In *Proc. Int'l Conf. On Automatic Face and Gesture Recognition*.

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

Wu, Y., and Huang, T. S. 1999. Capturing articulated human hand motion: A divide-and-conquer approach. In *Proc. IEEE Int'l Conf. on Computer Vision*, 606–611.

Wu, Y.; Liu, Q.; and Huang, T. S. 2000. An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization. In *Proc. of Asian Conference on Computer Vision*, 1106–1111.