Face Recognition by Support Vector Machines

Guodong Guo, Stan Z. Li, and Kapluk Chan
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798
{egdguo,eszli,eklchan}@ntu.edu.sg

Abstract

Support Vector Machines (SVMs) have been recently proposed as a new technique for pattern recognition. In this paper, the SVMs with a binary tree recognition strategy are used to tackle the face recognition problem. We illustrate the potential of SVMs on the Cambridge ORL face database, which consists of 400 images of 40 individuals, containing quite a high degree of variability in expression, pose, and facial details. We also present the recognition experiment on a larger face database of 1079 images of 137 individuals. We compare the SVMs based recognition with the standard eigenface approach using the Nearest Center Classification (NCC) criterion.

Keywords*: Face recognition, support vector machines, optimal separating hyperplane, binary tree, eigenface, principal component analysis.*

1 Introduction

Face recognition technology can be used in wide range of applications such as identity authentication, access control, and surveillance. Interests and research activities in face recognition have increased significantly over the past few years [12] [16] [2]. A face recognition system should be able to deal with various changes in face images. However, "the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity" [7]. This presents a great challenge to face recognition. Two issues are central, the first is what features to use to represent a face. A face image subjects to changes in viewpoint, illumination, and expression. An effective representation should be able to deal with possible changes. The second is how to classify a new face image using the chosen representation.

In geometric feature-based methods [12] [5] [1], facial features such as eyes, nose, mouth, and chin are detected. Properties and relations such as areas, distances, and angles, between the features are used as the descriptors of faces. Although being economical and efficient in achieving data reduction and insensitive to variations in illumination and viewpoint, this class of methods rely heavily on the extraction and measurement of facial features. Unfortunately, feature extraction and measurement techniques and algorithms developed to date have not been reliable enough to cater to this need [4].

In contrast, template matching and neural methods [16] [2] generally operate directly on an image-based representation of faces, *i.e.* pixel intensity array. Because the detection and measurement of geometric facial features are not required, this class of methods have been more practical and easy to implement as compared to geometric feature-based methods.

One of the most successful template matching methods is the eigenface method [15], which is based on the Karhunen-Loeve transform (KLT) or the principal component analysis (PCA) for the face representation and recognition. Every face image in the database is represented as a vector of weights, which is the projection of the face image to the basis in the eigenface space. Usually the nearest distance criterion is used for face recognition.

Support Vector Machines (SVMs) have been recently proposed by Vapnik and his co-workers [17] as a very effective method for general purpose pattern recognition. Intuitively, given a set of points belonging to two classes, a SVM finds the hyperplane that separates the largest possible fraction of points of the same class on the same side, while maximizing the distance from either class to the hyperplane. According to Vapnik [17], this hyperplane is called Optimal Separating Hyperplane (OSH) which minimizes the risk of misclassifying not only the examples in the training set but also the unseen examples of the test set.

The application of SVMs to computer vision problem have been proposed recently. Osuna *et al* [9] train a SVM for face detection, where the discrimination is between two classes: face and non-face, each with thousands of examples. Pontil and Verri [10] use the SVMs to recognize 3D objects which are obtained from the Columbia Object Image

Library (COIL) [8]. However, the appearances of these objects are explicitly different, and hence the discriminations between them are not too difficult. Roobaert *et al* [11] repeat the experiments again, and argue that even a simple matching algorithm can deliver nearly the same accuracy as SVMs. Thus, it seems that the advantage of using SVMs is not obvious.

It is difficult to discriminate or recognize different persons (hundrends or thousands) by their faces [6] because of the similarity of faces. In this research, we focus on the face recognition problem, and show that the discrimination functions learned by SVMs can give much higher recognition accuracy than the popular standard eigenface approach [15]. Eigenfaces are used to represent face images [15]. After the features are extracted, the discrimination functions between each pair are learned by SVMs. Then, the disjoint test set enters the system for recognition. We propose to construct a binary tree structure to recognize the testing samples. We present two sets of experiments. The first experiment is on the Cambridge Olivetti Research Lab (ORL) face database of 400 images of 40 individuals. The second is on a larger data set of 1079 images of 137 individuals, which consists of the database of Cambridge, Bern, Yale, Harvard, and our own.

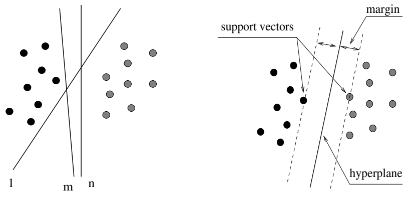
In Section 2, the basic theory of support vector machines is described. Then in Section 3, we present the face recognition experiments by SVMs and carry out comparisons with other approaches. The conclusion is given in Section 4.

2 Support Vector Machines for Pattern Recognition

2.1 Basic Theory of Support Vector Machines

For a two-class classification problem, the goal is to separate the two classes by a function which is induced from available examples. Consider the examples in Fig. 1 (a), where there are many possible linear classifiers that can separate the data, but there is only one (shown in Fig. 1 (b)) that maximizes the margin (the distance between the hyperplane and the nearest data point of each class). This linear classifier is termed the optimal separating hyperplane (OSH). Intuitively, we would expect this boundary to generalize well as opposed to the other possible boundaries shown in Fig. 1 (a).

Consider the problem of separating the set of training vectors belong to two separate classes, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, where $\mathbf{x}_i \in R^n$, $y_i \in \{-1, +1\}$ with a hyperplane $\mathbf{w}\mathbf{x} + b = 0$. The set of vectors is said to be *optimally separated* by the hyperplane if it is separated without error and the margin is maximal. A canonical hyperplane [17] has the constraint for parameters $\mathbf{w}$ and b: $\min_{\mathbf{x}_i} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$.



(a) (b)

Figure 1. Classification between two classes using hyperplanes: (a). arbitrary hyperplanes l, m and n; (b). the optimal separating hyperplane with the largest margin identified by the dashed lines, passing the two support vectors.

A separating hyperplane in canonical form must satisfy the following constraints,

$$y_i\left[(\mathbf{w} \cdot \mathbf{x}_i) + b\right] \geq 1, i = 1, \ldots, l \tag{1}$$

The distance of a point $\mathbf{x}$ from the hyperplane is,

$$d(\mathbf{w}, b; \mathbf{x}) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\parallel \mathbf{w} \parallel} \tag{2}$$

The margin is $\frac{2}{\|\mathbf{w}\|}$ according to its definition. Hence the hyperplane that optimally separates the data is the one that minimizes

$$\Phi(\mathbf{w}) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 \tag{3}$$

The solution to the optimization problem of (3) under the constraints of (1) is given by the saddle point of the Lagrange functional,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{l} \alpha_i \left\{y_i\left[(\mathbf{w} \cdot \mathbf{x}_i) + b\right] - 1\right\} \tag{4}$$

where α_i are the Lagrange multipliers. The Lagrangian has to be minimized with respect to $\mathbf{w}$, b and maximized with respect to $\alpha_i \geq 0$. Classical Lagrangian duality enables the *primal* problem (4) to be transformed to its *dual* problem, which is easier to solve. The *dual* problem is given by,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left\{ \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) \right\} \tag{5}$$

The solution to the *dual* problem is given by,

$$\bar{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (6)$$

with constraints,

$$\alpha_i \geq 0, \;\; i = 1, \ldots, l \quad (7)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (8)$$

Solving Equation (6) with constraints (7) and (8) determines the Lagrange multipliers, and the OSH is given by,

$$\bar{\mathbf{w}} = \sum_{i=1}^{l} \bar{\alpha}_i y_i \mathbf{x}_i \quad (9)$$

$$\bar{b} = -\frac{1}{2} \bar{\mathbf{w}} \cdot [\mathbf{x}_r + \mathbf{x}_s] \quad (10)$$

where $\mathbf{x}_r$ and $\mathbf{x}_s$ are support vectors, satisfying,

$$\bar{\alpha}_r, \bar{\alpha}_s > 0, \;\; y_r = 1, \;\; y_s = -1 \quad (11)$$

For a new data point $\mathbf{x}$, the classification is then,

$$f(\mathbf{x}) = sign\left(\bar{\mathbf{w}} \cdot \mathbf{x} + \bar{b}\right) \quad (12)$$

So far the discussion has been restricted to the case where the training data is linearly separable. To generalize the OSH to the non-separable case, slack variables ξ_i are introduced [3]. Hence the constraints of (1) are modified as

$$y_i \left[(\mathbf{w} \cdot \mathbf{x}_i) + b\right] \geq 1 - \xi_i, \;\; \xi_i \geq 0, \;\; i = 1, \ldots, l \quad (13)$$

The generalized OSH is determined by minimizing,

$$\Phi(\mathbf{w}, \xi) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{l} \xi_i \quad (14)$$

(where C is a given value) subject to the constraints of (13).

This optimization problem can also be transformed to its *dual* problem, and the solution is,

$$\bar{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (15)$$

with constraints,

$$0 \leq \alpha_i \leq C, \;\; i = 1, \ldots, l \quad (16)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (17)$$

The solution to this minimization problem is identical to the separable case except for a modification of the bounds of the Lagrange multipliers.

We only use the linear classifier in this research, so we do not further discuss the non-linear decision surfaces. See [17] for more about SVMs.

2.2 Multi-class Recognition

Previous subsection describes the basic theory of SVM for two class classification. A multi-class pattern recognition system can be obtained by combining two class SVMs. Usually there are two schemes for this purpose. One is the one-against-all strategy to classify between each class and all the remaining; The other is the one-against-one strategy to classify between each pair. While the former often leads to ambiguous classification [10], we adopt the latter one for our face recognition system.

We propose to construct a bottom-up binary tree for classification. Suppose there are eight classes in the data set, the decision tree is shown in Fig. 2, where the numbers 1-8 encode the classes. Note that the numbers encoding the classes are arbitrary without any means of ordering. By comparison between each pair, one class number is chosen representing the "winner" of the current two classes. The selected classes (from the lowest level of the binary tree) will come to the upper level for another round of tests. Finally, the unique class will appear on the top of the tree.

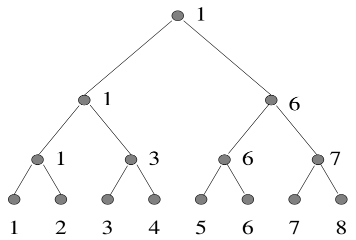


Figure 2. The binary tree structure for 8 classes face recognition. For a coming test face, it is compared with each two pairs, and the winner will be tested in an upper level until the top of the tree. The numbers 1-8 encode the classes. By bottom-up comparison of each pair, the unique class number will finally appear on the top of the tree.

Denote the number of classes as c, the SVMs learn $\frac{c(c-1)}{2}$ discrimination functions in the training stage, and carry out comparisons of $c-1$ times under the fixed binary tree structure. If c does not equal to the power of 2, we can decompose c as: $c = 2^{n_1}+2^{n_2}+\ldots+2^{n_I}$, where $n_1 \geq n_2 \geq \ldots \geq n_I$. Because any natural number (even or odd) can be decomposed into finite positive integers which are the power of 2. If c is an odd number, $n_I = 0$; if c is even, $n_I > 0$. Note that the decomposition is not unique, but the number of comparisons in the test stage is always $c-1$.

For example, given $c = 40$, we can decompose it as $40 = 32 + 8$. In testing stage, we do the tests firstly in the tree with 32 leaves and then another tree with 8 leaves. Finally, we compare these two outputs to determine the true class in another tree with only two leaves. The total number of comparisons for one query are 39.

3 Experimental Results

Two sets of experiments are presented to evaluate and compare the SVMs based algorithm with other recognition approaches.

3.1 Face Recognition on the ORL Face Database

The first experiment is performed on the Cambridge ORL face database, which contains 40 distinct persons. Each person has ten different images, taken at different times. We show four individuals (in four rows) in the ORL face images in Fig. 3. There are variations in facial expressions such as open/closed eyes, smiling/nonsmiling, and facial details such as glasses/no glasses. All the images were taken against a dark homogeneous background with the subjects in an up-right, frontal position, with tolerance for some side movements. There is also some variations in scale.

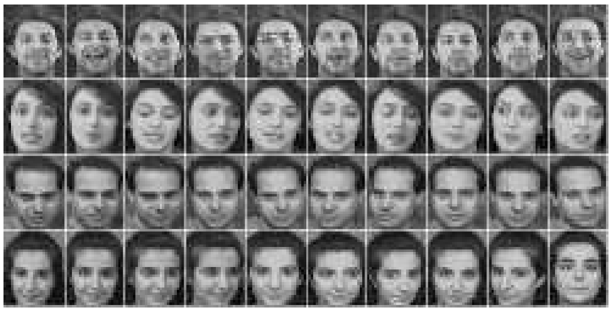


Figure 3. Four individuals (each in one row) in the ORL face database. There are 10 images for each person.

There are several approaches for classification of the ORL database images. In [14], a hidden Markov model (HMM) based approach is used, and the best model resulted in a 13% error rate. Later, Samaria extends the top-down HMM [14] with pseudo two-dimensional HMMs [13], and the error rate reduces to 5%. Lawrence *et al* [6] takes the convolutional neural network (CNN) approach for the classification of ORL database, and the best error rate reported is 3.83%(in the average of three runs).

In our face recognition experiments on the ORL database, we select 200 samples (5 for each individual) randomly as the training set, from which we calculate the eigenfaces and train the support vector machines (SVMs). The remaining 200 samples are used as the test set. Such procedures are repeated for four times, *i.e.*, four runs, which results in 4 groups of data. For each group, we calculate the error rates versus the number of eigenfaces (from 10-100). Figure 4 shows the results of the average of four runs. For comparison, we show the results of SVM and NCC [15] in the same figure. It is obvious that the error rates of SVM is much lower than that of NCC. The average minimum error rate of SVM in average is 3.0%, while the NCC is 5.25%. The minimum error rate of SVM in average is lower than the reported results 3.83% (in three runs) of CNN [6]. If we choose the best results among the four groups, the lowest error rate of the SVM can achieve 1.5%.

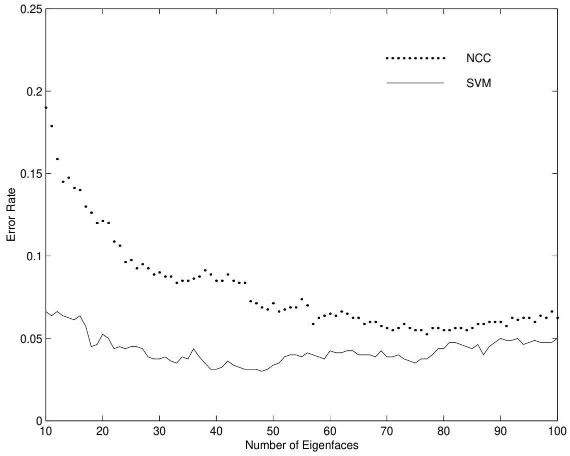


Figure 4. Comparison of error rates versus the number of eigenfaces of the standard NCC and SVM algorithms on the ORL face database.

3.2 Face Recognition on a Larger Compound Database

The second experiment is performed on a compound data set of 1079 face images of 137 persons, which consists of five databases:

(1). The Cambridge ORL face database described previously. (2). The Bern database contains frontal views of

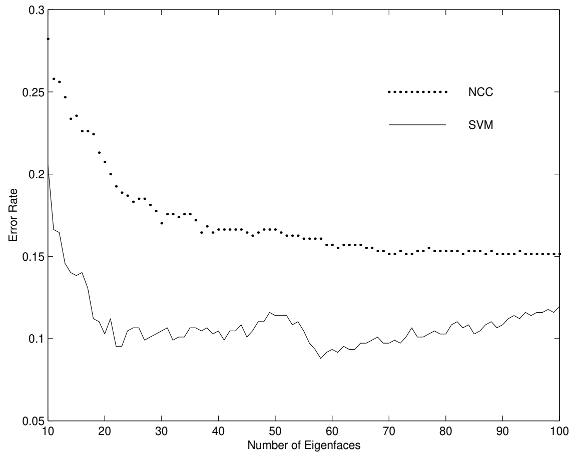


Figure 5. Comparison of error rates versus the number of eigenfaces of the standard NCC and SVM algorithms on the compound face database.

30 persons. (3). The Yale database contains 15 persons. For each person, ten of its 11 frontal view images are randomly selected. (4). Five persons are selected from the Harvard database. (5). A database of our own, composed of 179 frontal views of 47 Chinese students, each person having three or four images taken at different facial expression, viewpoints and facial details.

A subset of the compound data set is used as the training set for computing the eigenfaces, and learning the discrimination functions by SVMs. It is composed of 544 images: five images per person are randomly chosen from the Cambridge, Bern, Yale, and Harvard databases, and two images per person are randomly chosen from our own database. The remaining 535 images are used as the test set.

In this experiment, the number of classes $c = 137$, and the SVMs based methods are trained for $\frac{c(c-1)}{2} = 18632$ pairs. To construct the binary trees for testing, we decompose $137 = 32 + 32 + 32 + 32 + 8 + 1$. So we have 4 binary trees each with 32 leaves, denoted as T_1, T_2, T_3, and T_4, respectively, and one binary tree with 8 leaves, denoted as T_5, and one class is left, coded as lc. The 4 classes appear at the top of T_1, T_2, T_3, and T_4 are used to construct another 4-leaf binary tree T_6. The outputs of T_5 and T_6 construct a 2-leaf binary tree T_7. Finally, the output of T_7 and the left class lc will construct another 2-leaf tree T_8. The true class will appear at the top of T_8.

For each query, the SVMs need testing for 136 times. Although the number of comparisons seem high, the process is fast, as each test just computes an inner product and only uses its sign.

Our construction of the binary decision trees has some similarity to the "tennis tournament" proposed by Pontil and Verri [10] in their 3D object recognition. However, they assume there are 2^K players, and they just select 32 objects from 100 in the COIL images [8]. They do not address the problem when an arbitrary number of objects are encountered. Through the construction of several binary trees, we can solve a recognition problem with any number of classes.

We compare SVMs with the standard eigenface method [15] which takes the nearest center classification (NCC) criterion. Both approaches start with the eigenface features, but different in the classification algorithm. The error rates are calculated as the function of the number of eigenfaces, *i.e.*, the feature dimensions. We display the results in Fig. 5. The minimum error rate of SVM is 8.79%, which is much better than the 15.14% of NCC.

4 Conclusions

We have presented the face recognition experiments using linear support vector machines with a binary tree classification strategy. As shown in the comparison with other techniques, it appears that the SVMs can be effectively trained for face recognition. The experimental results show that the SVMs are a better learning algorithm than the nearest center approach for face recognition.

References

[1] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1042–1052, 1993.

[2] R. Chellappa, C. L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proc. IEEE*, 83:705–741, May 1995.

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

[4] I. J. Cox, J. Ghosn, and P. Yianilos. Feature-based face recognition using mixture-distance. *CVPR*, pages 209–216, 1996.

[5] A. J. Goldstein, L. D. Harmon, and A. B. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, May 1971.

[6] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural network approach. *IEEE Trans. Neural Networks*, 8:98–113, 1997.

[7] Y. Moses, Y. Adini, and S. Ullman. Face recognition: the problem of compensating for changes in illumination direction. *European Conf. Computer Vision*, pages 286–296, 1994.

[8] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *Int. Journal of Computer Vision*, 14:5–24, 1995.

[9] E. Osuna, R. Freund, and F. girosi. Training support vector machines: an application to face detection. *Proc. CVPR*, 1997.

[10] M. Pontil and A. Verri. Support vector machines for 3-d object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:637–646, 1998.

[11] D. Roobaert, P. Nillius, and J. Eklundh. Comparison of learning aproaches to appearance-based 3d object recognition with and without cluttered background. *ACCV2000*, to appear.

[12] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25:65–77, 1992.

[13] F. S. Samaria. *Face recognition using Hidden Markov Models*. PhD thesis, Trinity College, University of Cambridge, Cambridge, 1994.

[14] F. S. Samaria and A. C. Harter. Parameterization of a stochastic model for human face identification. *Proceedings of the 2nd IEEE workshop on Applications of Computer Vision*, 1994.

[15] M. A. Turk and A. P. Pentland. Eigenfaces for recognition. *J. Cognitive Neurosci.*, 3(1):71–86, 1991.

[16] D. Valentin, H. Abdi., A. J. O'Toole, and G. W. Cottrell. Connectionist models of face processing: A survey. *Pattern Recognition*, 27:1209–1230, 1994.

[17] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.