Bayesian Face Recognition Using Support Vector Machine and Face Clustering

Zhifeng Li and Xiaoou Tang

Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{zli0, xtang}@ie.cuhk.edu.hk

Abstract

In this paper, we first develop a direct Bayesian based Support Vector Machine by combining the Bayesian analysis with the SVM. Unlike traditional SVM-based face recognition method that needs to train a large number of SVMs, the direct Bayesian SVM needs only one SVM trained to classify the face difference between intra-personal variation and extra-personal variation. However, the added simplicity means that the method has to separate two complex subspaces by one hyper-plane thus affects the recognition accuracy. In order to improve the recognition performance we develop three more Bayesian based SVMs, including the one-versus-all method, the Hierarchical Agglomerative Clustering based method, and the adaptive clustering method. We show the improvement of the new algorithms over traditional subspace methods through experiments on two face databases, the FERET database and the XM2VTS database.

1. Introduction

A number of subspace based face recognition methods have been developed in recent years. Eigenface method [1] uses the Karhunen-Loeve Transform (KLT) to produce a most expressive subspace for face representation and recognition. LDA or Fisherface [2][3][4] uses linear discriminant analysis to seek a set of features best separating face classes. Another important subspace method is Bayesian algorithm using probabilistic subspace [5]. Different from other subspace techniques, which classify the test face image into *M* classes of *M* individuals, the Bayesian algorithm casts the face recognition problem into a binary pattern classification problem with each of the two classes, intrapersonal variation and extrapersonal variation, modeled by a Gaussian distribution.

After subspace features are computed, most methods use simple Euclidian distance of the subspace features to classify the face images. Recently, more sophisticated classifiers, such as support vector machines (SVM) [6], have been shown to be able to further improve the classification performance of the PCA and LDA subspace features [7][8][9]. Given any two classes of vectors, the aim of support vector machines is to find one hyperplane to separate the two classes of vectors so that the distance from the hyperplane to the closest vectors of both classes is the maximum. The hyperplane is known as the optimal separating hyperplane. Support vector machines excel at two-class recognition problem and outperform many other linear and non-linear classifiers.

Since SVM is basically a binary classifier, to apply it to face recognition, which is a typical multi-class recognition problem, we have to reduce the multi-class classification to a combination of SVMs. There are several strategies to solve this problem, among which one-versus-all strategy and pairwise strategy are often used [7][10]. Although both approaches can achieve high recognition accuracy, the former is much simpler than the latter. Studies have shown similar face classification performance for the two approaches [7].

Since the number of classes in face recognition is often very large, for both the one-versus-all strategy and the pairwise strategy, a large number of SVMs have to be trained. In order to alleviate this problem, besides a one-versus-all Bayesian SVM algorithm, we also develop a direct Bayesian SVM by combining the Bayesian analysis method with the SVM directly. The Bayesian method effectively converts the multi-class face recognition problem into a two-class classification problem, which is suitable for using the SVM directly. Therefore, the Bayesian SVM needs only one SVM trained to classify the face difference between intra-personal variation and extra-personal variation.

However, using only one hyper-plane may not be enough to separate the entire within-class space and between-class space given the large number of samples. From experimental comparison we see that the simplicity of the direct Bayesian SVM comes at a cost of accuracy. We can see that the two methods are at the two extremes, one needs too many classifiers and the other has too few classifiers. In order to balance the two extremes, we further develop a two-stage Bayesian SVM. In the first stage we estimate a similarity matrix to measure the degree of similarity between each pair of faces using the

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)
1063-6919/04 $20.00 © 2004 **IEEE**

direct Bayesian SVM. Then using the similarity matrix and the Hierarchical Agglomerative Clustering (HAC) algorithm [11][12] we group all face classes into clusters of similar faces. In the second stage we perform the one-versus-all SVMs on the small number of classes within each cluster. During testing, we first use the original Bayesian method to classify the probe face to a cluster, and then the final classification is obtained within this cluster by the second stage SVM. The method is shown to be as effective as the one-versus-all approach but is more efficient in computation.

Notice that the clustering is based on the training data thus stays the same in the testing stage. In order to cluster the data adaptively for each testing face we develop an adaptive clustering Bayesian SVM algorithm. We first use a simple Bayesian algorithm to find a cluster of faces that are most similar to the testing face, then use a one-versus-all algorithm to re-classify the face in this cluster to find the final result. Finally, we apply the adaptive clustering algorithms to the unified subspace analysis method [15] to further improve the classification performance. We use experiments on two face databases, the FERET face database [13] and the XM2VTS face database [14] to compare the new algorithms with traditional subspace methods.

2. Combining Bayesian and Support Vector Machine

In this section, we first briefly review the support vector machine [6] and Bayesian face recognition [5]. We then develop the direct Bayesian SVM and the one-versus-all Bayesian SVM.

2.1. Support Vector Machines

Consider *N* points that belong to two different classes,

$$\{(x_i, y_i)\}_{i=1}^{N} \text{ and } y_i = \{+1, -1\}, \tag{1}$$

where x_i is an *n*-dimension vector and y_i is the label of the class that the vector belongs to. SVM separates the two classes of points by a hyper-plane,

$$w^T x + b = 0, \tag{2}$$

where x is an input vector, w is an adaptive weight vector, and b is a bias. The functional margin of the hyper-plane is represented as,

$$\begin{cases} w_0^T x_i + b_0 \geq +1 & y_i = +1 \\ w_0^T x_i + b_0 \leq -1 & y_i = -1 \end{cases}. \tag{3}$$

For a given w_0 and b_0, the geometrical distance of a point *x* from the optimal hyper-plane is,

$$d(w_0, b_0, x) = \frac{|w_0 x + b_0|}{\|w_0\|}. \tag{4}$$

The goal of the SVM is to find the parameters w_0 and b_0 for the optimal separating hyper-plane to maximize the geometrical margin, $\frac{2}{\|w\|}$, i.e. the distance between the hyper-plane and the closest point of both classes. Hence the hyper-plane that optimally separates the data is the one that minimizes

$$\phi(w) = \frac{1}{2}\|w\|_2^2 = \frac{1}{2}(w \cdot w'), \tag{5}$$

subject to the constraints $y_i(w \cdot x_i + b) \geq 1, \forall i$. The solution to this optimization problem is found through the maximization of the dual Lagrangian,

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j), \tag{6}$$

with respect to Lagrange multiplier α_i, subject to the constraints,

$$\alpha_i \geq 0, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{7}$$

In the solution, only a small number of α_i is none zero, each of which corresponds to one training data point. These data points are called support vectors since they lie on the margin border. These support vectors are therefore the only data points that appear in the resulting hyper-plane, i.e. decision function,

$$f_l(x) = \sum_{i=1}^{m} y_i \alpha_i \langle x, x_{si} \rangle + b, \tag{8}$$

where each x_{si} represents a support vector and *m* is the number of support vectors. Each test vector *x* is then classified by the sign of *f(x)*.

The solution can be extended to the case of nonlinear separating hyper-planes by a mapping of the input space into a high dimensional space, $x \to \phi(x)$. The key property of this mapping is that the function ϕ is subject to the condition that the dot product of the two functions $\phi(x_i) \cdot \phi(x_j)$ can be rewritten as a kernel function $K(x_i, x_j)$. The decision function in Eq. (8) then becomes,

$$f(x) = \sum_{i=1}^{m} y_i \alpha_i K(x, x_{si}) + b\,, \tag{9}$$

We use the popular Gaussian kernel in our study.

2.2. Bayesian Analysis

The Bayesian analysis converts the multi-class face recognition problem into a two-class classification problem by classifying the face difference as intra-personal variations for the same person and extra-personal variations for different persons [5]. Let Ω_I represent the intra-personal variations and Ω_E represent the extra-personal variations, the ML similarity between any two images can be defined as,

$$S(I_1, I_2) = P(\Delta \mid \Omega_I) \quad . \tag{10}$$

To estimate $P(\Delta \mid \Omega_I)$, we perform PCA on the face difference set $\{\Delta \mid \Delta \in \Omega_I\}$ to decompose the image difference space into two orthogonal and complementary subspaces: the principle subspace F, called intra-personal eigenspace with K eigenvectors, and its complementary space $\overline{F}$ with $N - K$ eigenvectors. The likelihood can be estimated as,

$$\hat{P}(\Delta \mid \Omega_I) = \left[\frac{\exp\left(-\frac{1}{2} d_F(\Delta)\right)}{(2\pi)^{K/2} \prod_{i=1}^{K} \lambda_i^{1/2}} \right] \left[\frac{\exp\left(-\varepsilon^2(\Delta)/2\rho\right)}{(2\pi\rho)^{(N-K)/2}} \right], \tag{11}$$

where, $d_F(\Delta)$ is the so-called distance-in-feature-space (DIFS),

$$d_F(\Delta) = \sum_{i=1}^{K} \frac{y_i^2}{\lambda_i}\,. \tag{12}$$

In Eq. (11) and (12), y_i is the principle component of the principle subspace F, λ_i is the corresponding eigenvalue, $\varepsilon^2(\Delta)$ is the PCA residual error in $\overline{F}$, also called the "distance-from-feature-space" (DFFS), and ρ is the average of all the eigenvalues of $\overline{F}$,

$$\rho = \frac{1}{N-K} \sum_{i=K+1}^{N} \lambda_i\,. \tag{13}$$

From Eq. (11), we can see that the estimation of $P(\Delta \mid \Omega_I)$ is equivalent to computing the distance measure in the intrapersonal subspace,

$$D_I = d_F(\Delta) + \varepsilon^2(\Delta)/\rho\,. \tag{14}$$

We use DIFS in our study since DFFS and ML are much more costly to compute.

2.3. Combining Bayesian and SVM

As discussed before, SVM is a binary classifier. For face recognition problem we need to extend it to a multi-class classifier. The pair-wise strategy and the one-versus-all strategy are the two most popular methods. For the pair-wise strategy, one support vector machine is trained to separate each pair of classes. So the method needs c*(c-1)/2 support vector machines trained, where c is the number of classes. During the testing, each support vector machine votes for one class, and the winning class is the one that has the largest number of votes. For the one-versus-all strategy, each support vector machine is trained to separate a single class from the remaining classes. In other words, each class is associated to one hyper-plane. So it needs c support vector machines trained. Each test vector is assigned to the class whose hyper-plane is farthest from it. Since the one-versus-all method is simpler and is as effective as the pair-wise method, we first adopt it to implement a straightforward one-versus-all Bayesian SVM.

However, for face recognition, the number of classes often is very large. The one-versus-all method needs to train a large number of SVMs. In order to alleviate this problem, we develop a direct Bayesian SVM for face classification. The method is straightforward since the traditional Bayesian algorithm already converts the face recognition problem into a two-class problem for the intra-personal and the extra-personal variation. We therefore only need to train one SVM for the two-class features.

For the training data, we first compute image difference between images of the same person to construct the intrapersonal variation set $\{\Delta_I \mid \Delta_I \in \Omega_I\}$. We then compute image difference between images of different persons to construct the extra-personal variation set $\{\Delta_E \mid \Delta_E \in \Omega_E\}$. The eigenvalue matrix Λ_I and eigenvector matrix V_I of the intra-personal subspace are then computed from the intra-personal variation set $\{\Delta_I \mid \Delta_I \in \Omega_I\}$. Finally, all the image difference vectors are projected and whitened in the intra-personal subspace,

$$\Delta_I' = \Lambda_I^{-\frac{1}{2}} V_I^T \Delta_I\,. \tag{15}$$

$$\Delta_E' = \Lambda_I^{-\frac{1}{2}} V_I^T \Delta_E\,. \tag{16}$$

These two sets of image difference vectors are used to train the SVM to generate the decision function $f(\Delta)$.

For the testing process, we again compute the face difference vector Δ_i between the probe vector x and each gallery vector x_i^g, and then project and whiten the difference vector in the intra-personal subspace,

$$\Delta_i' = \Lambda_I^{-\frac{1}{2}} V_I^T \Delta_i . \quad (17)$$

The final classification decision is made by,

$$d(x) = \arg\max_{1 \le i \le c} \left(f\left(\Delta_i'\right) \right). \quad (18)$$

where c is the number of people in the gallery. The larger is the value of d, the more reliable the result is.

The direct Bayesian SVM is simpler than the one-versus-all Bayesian SVM since it only needs one SVM trained. However, this new method may have over simplified the problem since it uses one hyper-plane to separate the intra-personal variation and the extra-personal variation. To balance the trade off between the two methods, we develop a two-stage SVM method in the next section.

3. Two-Stage Clustering Based Classification

The problem with the one-versus-all approach is that too many SVMs need to be trained. On the contrary, the problem with the direct Bayesian SVM is too many samples for just one SVM. In this section, we try to find a solution that balances the two extremes.

When we train a SVM, the most important region in the training data space is around the decision hyper-plane, since that is where mistakes often happen. Samples that are further away from the hyper-plane play less significant roles in the training process. Therefore it is reasonable to train a SVM for samples that are near the hyper-plane. Toward this, we first partition the gallery data into clusters, with each cluster containing only similar images.

We first use the Bayesian SVM to quickly estimate the similarity matrix of the gallery set, and then use the Hierarchical Agglomerative Clustering (HAC) technique [11][12] to group the similar face clusters in order to reduce the number of binary SVMs in the second stage.

3.1. Hierarchical Agglomerative Clustering (HAC)

In the Hierarchical Agglomerative Clustering process, clusters are constructed by combining existing clusters based on their proximity. The basic process of the HAC can be summarized by the following steps:

(1) Initialize a set of clusters.

(2) Find the nearest pair of clusters that have the largest similarity measure, and then merge them into a new cluster. Estimate the similarity measure between the new cluster and all the other clusters.

(3) Repeat step (2) until it satisfies the stopping rule.

In each of the three steps of the basic algorithm, different strategies can be used to lead to different designs of the HAC algorithm. For example, in the first step, we can either assign each data point as a distinct cluster or form some initial small clusters for seeding. For face recognition, we can simply assign each image in the gallery as a cluster (assuming only one image per person in the gallery). In the third step, the stopping rule could either be that clustering has reached its root, or the clustering has reached the number of clusters specified by the user, or the similarity measure between the two nearest clusters is above a preset threshold. In our study, we will use the cluster number as stopping criteria. One of the key design issues for the HAC algorithm is the similarity measure between clusters in the second step. In the new algorithm described in the following section, we use the direct Bayesian SVM to estimate the similarity measure between face clusters. The output of the HAC will be a dendrogram. An example is shown in Fig. 1, where 10 classes are merged into 3 clusters.

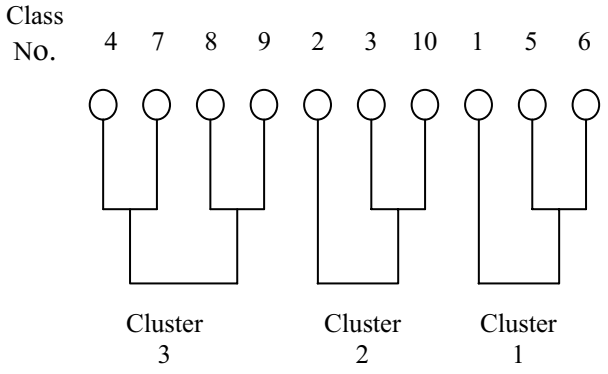


Figure 1. A dendrogram example.

3.2. Two-Stage SVM

In order to use HAC to partition the gallery face data into clusters, we first need to compute the similarity among the face images. For a pair of face images i and j in the gallery, we first compute the image difference Δ_E^{ij}, then project and whiten it in the intra-personal subspace,

$$\Delta_E^{ij\,'} = \Lambda_I^{-\frac{1}{2}} V_I^T \Delta_E^{ij} . \quad (19)$$

The similarity measure between the two images is then defined as,

$$S_{ij} = f(\Delta_E^{ij\,'}) , \quad (20)$$

where f is the SVM decision function in Eq. (9). The further away is the image difference from the decision hyper-plane, the larger the similarity value is. This means the image difference is closer to intra-personal variation, thus the two images are more similar to each other. The similarity values for all the image pairs form the similarity matrix for the image gallery set.

Using the similarity matrix, we then group the gallery data set into clusters of similar images through the HAC.

After the similar images are clustered, in the second stage, we perform the one-versus-all Bayesian SVM within each cluster. Since the image number is much smaller in each cluster, the training complexity is significantly reduced. In addition, the SVM only need to focus on a small number of similar images within each cluster. These data points are closer to the decision surface, thus are more likely to become support vectors.

During the testing, we first compute the whitened face difference vector Δ_i' between the probe vector x and each gallery vector, and then simply find the face class that gives the smallest Δ_i'. This is equivalent to the original Bayesian method. If the output is class k, we find the face cluster $C(k)$ that contains class k. A second stage one-versus-all SVM is then performed on the cluster $C(k)$ to obtain the final classification result. Since the original Bayesian method only requires computation between two short feature vectors so it is much faster and is used in the first stage to rank all the data. Then the more costly one-versus-all Bayesian SVM is only needed to process one small cluster. So the complexity of the HAC clustering based algorithm is much less than the one-versus-all approach.

However, since the clustering is based on the training data only, the face clusters will stay the same in the testing stage. They are tuned to the training data without any adaptation to the testing data. In order to cluster the data adaptively for each testing face we further develop an adaptive clustering Bayesian SVM algorithm. We first use the original Bayesian algorithm to find a cluster of faces that are most similar to the testing face. We then use a one-versus-all algorithm to re-classify the face in this cluster to find the final result. Unlike the HAC clustering approach that only need to train SVM classifiers in the training stage, if we have to re-train the one-versus-all classifier for each new cluster in the testing stage, the cost of computation will be simply too high. Instead, we train the one-versus-all Bayesian SVM in the training stage for all the training data just like the original one-versus-all Bayesian SVM. We then use this one-versus-all Bayesian SVM to re-classify only the faces in the new cluster. So for training the complexity is the same as one-versus-all, but for testing, the new cluster method will be much fast since it only need to focus on a small cluster and the first step original Bayesian algorithm is much faster. In experiments, we will see that this algorithm improves the recognition accuracy over all other methods.

So far, we have been focusing on Bayesian face recognition. In fact, the two-stage SVM can also be extended to other subspace methods. The unified subspace analysis method [15] has been shown to outperform most of the traditional subspace methods. Here, we apply the two-stage cluster based SVM to the unified subspace method. Experiments show that this method achieves even better results than the Bayesian based SVM methods.

4. Experiments

In this section, we conduct experiments on two face databases, the FERET face database [13] and the XM2VTS face database [14]. To better evaluate the recognition performance we preprocess the face images through the following steps: 1) rotate the face images to align the vertical face orientation; 2) scale the face images so that the distances between the two eyes are the same for all images; 3) crop the face images to remove the background and the hair region; 4) apply histogram equalization to the face images for photometric normalization.

4.1. Experiment on the FERET face database

For the FERET face database (fa/fb), we use 495*2 images of 495 people as training data, and use images of the other 700 people as testing data. Therefore, the gallery set is composed of 700 images of 700 people. The probe set is composed of 700 images of the same 700 people.

The recognition results of all the tested methods are summarized in Table 1. From the results we can see that the direct Bayesian SVM is only slightly better than the original Bayesian algorithm. This lack of significant improvement confirms that using only one hyperplane is not enough to separate the intrapersonal and extrapersonal subspaces. Both the one-versus-all method and the HAC based method improve the recognition accuracy significantly. Comparing to original Bayesian method, the recognition error rate is reduced by 45%. Finally, the adaptive clustering method gives the best accuracy of 97.4% among all Bayesian based methods. This is a very high accuracy for the FERET database. Both the HAC clustering and adaptive clustering methods are more efficient in computational cost since they only need to compute a small number of SVMs in the testing stage.

The good result for the adaptive clustering method is particularly interesting. Given that we use a regular one-versus-all method to re-classify the cluster of images selected by the original Bayesian method in the first step, instead of re-train the SVM, the method is effectively the

same as combining the two classifier in a series operation. For the sake of comparison, we can also use the one-versus-all method first then use the original Bayesian method. Of course, this is not a good approach since the former method is more expensive to compute. We select different number of samples in the first step clustering for the two methods and compute the recognition accuracy. Figure 2(a) shows the results for both methods. Clearly, using the first approach is much better.

This can be explained by the complementary properties of the two classifiers. The Bayesian method is more stable but less accurate. The one-versus-all Bayesian on the other hand is more accurate but less stable, since it is possible that one or a few of the large number of SVMs may produce a larger than normal distance measure outlier that happens to over shadow the real face class. When a stable Bayesian classifier is used first, it will help to remove these outliers from the selected cluster of candidates to help to improve the performance of the one-versus-all Bayesian classifier. In the experiment, the algorithm reaches the best performance with only 20 images in the cluster. If we use the less stable one-versus-all method first then use the original Bayesian, the performance is actually worse than using one-versus-all method alone, since the Bayesian method is less accurate. As the number of the images in the cluster increase, the combined method actually gets closer to the second algorithm with decreased influence of the first.

Finally, when using the adaptive clustering method on the unified subspace method, the recognition error rate is further reduced by 45%. We achieve the best accuracy of 98.6% on the FERET database.

4.2. Experiment on the XM2VTS face database

For the *XM2VST* database, we select all 295 people with four face images from four different sessions for each person. For the training data, we select 295*3 images of 295 people from the first three sessions. The gallery set is composed of 295 images of 295 people from the first session. The probe set is composed of 295 images of 295 people from the fourth session.

We implement the comparative experiments similar to the FERET face database experiment. Although the data size is smaller than the FERET database, the fact that the probe set and the gallery set in this experiment are from different sessions makes the recognition task also very challenging. This can be seen from the poor results of the PCA method, which is similar to direct matching of face images. The recognition results of all the tested methods are summarized in Table 1. The adaptive clustering recognition results for different number of images are shown in Fig. 2(b). The results further confirm our observation in the FERET data experiments.

Table 1. Recognition error rate on the FERET database and the XM2VTS database.

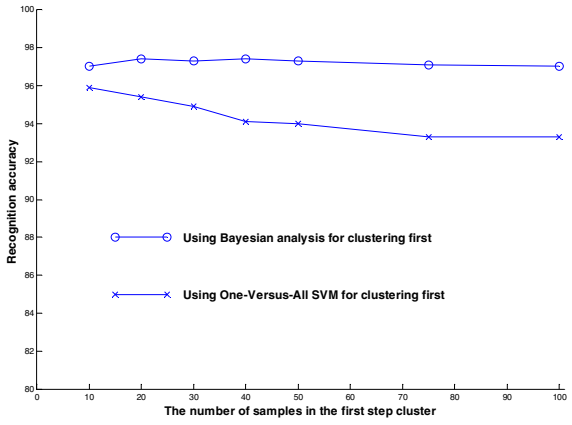
Methods	Recognition error rate (%)	
	FERET	XM2VTS
PCA	15.4	33.9
LDA	9.7	11.9
Bayesian	6.7	11.5
Unified Subspace	3.9	6.8
Direct Bayesian SVM	6.1	10.8
One-Versus-All Bayesian SVM	4.0	2.7
HAC Bayesian SVM	3.7	2.7
Adaptive Clustering Bayesian SVM	2.6	1.0
Adaptive Clustering Unified Subspace SVM	1.4	1.0

5. Conclusion

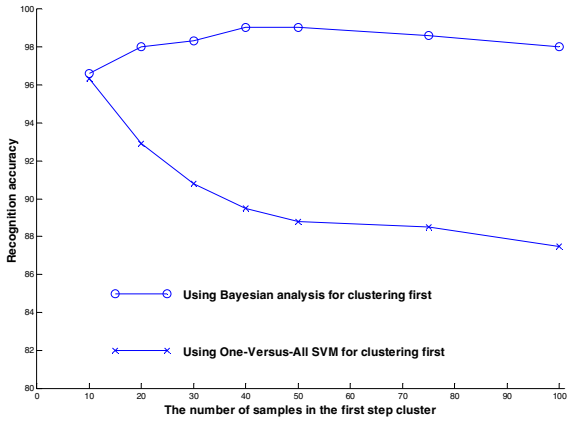
In this paper, we first develop a direct Bayesian based Support Vector Machine by combining the Bayesian analysis with the SVM. The direct Bayesian SVM needs only one SVM trained to classify the face difference between within-class variation and between-class variation. However, with the added simplicity, the new method also has an inherent drawback. It tries to separate two complex subspaces by just one hyperplane. In order to improve the recognition performance we further develop three more Bayesian based SVMs, including the one-versus-all method, the HAC based method, and the adaptive clustering method. Experimental results clearly demonstrate the superiority of the new algorithm over traditional subspace methods. In addition, the clustering strategy is also extended to the unified subspace face recognition method.

Finally, as pointed out earlier, similar to traditional subspace methods, all the new Bayesian based SVM methods developed in this paper can be easily applied to

local features such as Elastic graph Gabor features or Active Shape Model local features to further improve the recognition performance.



(a) FERET database.



(b) XM2VTS database.

Figure 2. Comparison of the recognition results for adaptive clustering using different number of samples in the first step cluster.

Acknowledgement

The work described in this paper was fully supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK 4190/01E and CUHK 4224/03E).

References

[1] M. Turk and A. Pentland "Face recognition using eigenfaces," in *Proc. of IEEE International Conference Computer Vision and Pattern Recognition*, pp. 586-591, 1991.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenface vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7): 711-720, 1997.

[3] W. Zhao and R. Chellappa, "Discriminant analysis of principal components for face recognition," *IEEE Conf. Automatic Face and Gesture Recognition*, Page(s): 336 –341, 1998.

[4] K. Etemad and R. Chellappa, "Face recognition using discriminant eigenvectors," *Proc. ICASSP*, Vol. 4, pp. 2148-2151, 1996.

[5] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, Vol. 33, pp. 1771-1782, 2000.

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

[7] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach," *Proceeding of ICCV*, Vol. 2, pp. 688-694, 2001.

[8] G. Guo, S. Z. Li, and K. Chan, "Face recognition by support vector machines," *Proceeding of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 196-201, 2000.

[9] D. Xi, I. T. Podolak, and S. Lee, "Facial component extraction and face recognition with support vector machines," *Proceeding of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 76-81, 2002.

[10] E. Allwein, R. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifier," *Proceeding of ICML*, 2000.

[11] R. Duda, and P. Hart, *Pattern classification and scene analysis*, New York: Wiley, 1973.

[12] R. Yager, "Intelligent control of the hierarchical agglomerative clustering process," *IEEE Transaction on System, Man, and Cybernetics – Part B: Cybernetics*, Vol. 30, No. 6, December 2000.

[13] P. Phillips, H. Moon, P. Rauss, and S. Rizvi, "The FERET evaluation methodology for face-recognition algorithms," *Proc. of IEEE CVPR*, pp. 137-143, 1997.

[14] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Matitre, "XM2VTSDB: The extended M2VTS database," *Second International Conference on Audio and Video-based Biometric Person Authentication*, March 1999.

[15] X. Wang and X. Tang, "Unified subspace analysis for face recognition," *Proceeding of IEEE International Conference on Computer Vision*, 2003.