Locally Adaptive Classification Piloted by Uncertainty

Juan Dai JDAI5@IE.CUHK.EDU.HK
Shuicheng Yan SCYAN@IE.CUHK.EDU.HK
Department of Information Engineering, Chinese University of Hong Kong, Shatin, Hong Kong

Xiaoou Tang XITANG@MICROSOFT.COM
Microsoft Research Asia, Beijing, China

James T. Kwok JAMESK@CS.UST.HK
Department of Computer Science, Hong Kong University of Science and Technology, Kowloon, Hong Kong

Abstract

Locally adaptive classifiers are usually superior to the use of a single global classifier. However, there are two major problems in designing locally adaptive classifiers. First, how to place the local classifiers, and, second, how to combine them together. In this paper, instead of placing the classifiers based on the data distribution only, we propose a *responsibility mixture model* that uses the uncertainty associated with the classification at each training sample. Using this model, the local classifiers are placed near the decision boundary where they are most effective. A set of local classifiers are then learned to form a global classifier by maximizing an estimate of the probability that the samples will be correctly classified with a nearest neighbor classifier. Experimental results on both artificial and real-world data sets demonstrate its superiority over traditional algorithms.

1. Introduction

Linear models have been very popular because of their simplicity and analytical tractability, and algorithms like Eigenfaces (PCA) (Turk & Pentland, 1991) and linear discriminant analysis (LDA) (Belhumeur et al., 1997) have been widely used in many real-world applications. However, when the data has a complex nonlinear structure, a single linear classifier cannot well separate the different classes. A natural remedy is then to have an ensemble of locally adaptive classifiers. There are two fundamental problems in designing such an ensemble:

1. How to place the local classifiers?
2. How to effectively fuse these local classifiers?

Attempts have been made in solving these problems (Kim & Kittler, 1994; Yan et al., 2004; Meir et al., 2000; Chapelle et al., 2000; Toussaint & Vijayakumar, 2005). Kim and Kittler (1994) place the local classifiers at the clusters obtained by the K-means (Selim & Ismail, 1984) clustering algorithm. The projection directions of the local classifiers are then derived by using the (global) Fisher criterion (Fisher, 1936). Yan et al. (2004) proposed a variant of the K-means algorithm called *Intra-Class Balanced K-means.* The local classifiers are placed in clusters such that each one has a balanced number of samples from the different classes. They then also use the Fisher criterion to optimize the projection directions of the classifiers. The work of Kim and Kittler (1994) can easily decide the locations of the local classifiers since the method is unsupervised and based directly on the data distribution. The work of Yan et al. (2004), on the other hand, can utilize the label information. However, it becomes computationally infeasible especially when the classes are imbalanced.

In this paper, we propose a different approach for solving these two problems. To address the first problem, the local classifiers are placed by using an *uncertainty map*, where uncertainty refers to the probability that a sample will be misclassified by the nearest neighbor classifier. A *responsibility mixture model* is learned to describe this uncertainty distribution. The local classifiers are then placed in areas with high uncertainties.

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

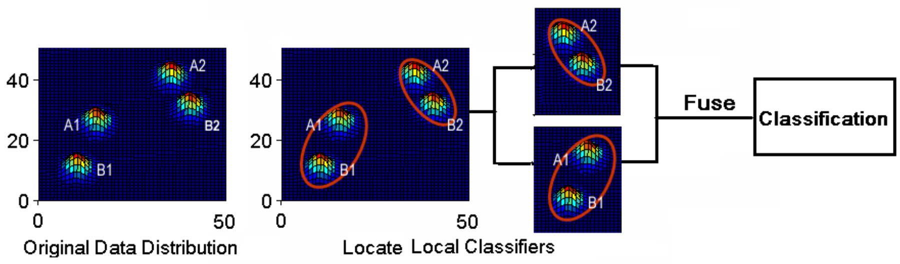


Figure 1. Schematic diagram for designing an ensemble of locally adaptive classifiers. In this example, the data distributions of classes A and B are both Gaussian mixtures ($A1, A2$ are the two mixture components for class A, while $B1, B2$ are those for class B). The two ellipses show the locations of the local classifiers.

For the second problem, we introduce a linear classifier with local dimensionality reduction that directly maximizes an estimate of the classification accuracy attained by a nearest neighbor classifier. The general design for an ensemble of locally adaptive classifiers is shown schematically in Figure 1.

The rest of this paper is organized as follow. Section 2 introduces the responsibility mixture model together with the uncertainty map. Section 3 then introduces the fusion of the local classifiers. Experimental results are presented in Section 4, and the last section gives the conclusion and future works.

2. Placement of the Local Classifiers

In a classification problem, we are given a training set of N i.i.d. samples. This can be represented by the matrix $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{m\times N}$ with $x_i \in \mathbb{R}^m$. The label of x_i is $c_i \in \{1, 2, \ldots, N_c\}$, where N_c is the number of classes. We also denote the number of samples belonging to the cth class by n_c, and the corresponding index set of samples by π_c.

Despite the possibly complicated global structure of the data, we can often represent it by a collection of simpler, locally linear models (Roweis & Saul, 2000; Bregler & Omohundro, 1995). Methods such as the Gaussian Mixture Model (GMM) (Bilmes, 1998) and Mixture of Factor Analyzers (MFA) (Ghahramani & Hinton, 1996) have been proposed for isolating these local structures. However, as these are unsupervised learning techniques, they may not be suitable in classification problems. This deficiency can be easily demonstrated by interchanging the labels of two samples belonging to different classes. This will not alter the clustering result, but the decision boundary can be changed significantly. Moreover, it is also possible that all the samples in one cluster may belong to the same class, and the classifier located at that cluster is thus unable to learn.

2.1. The Uncertainty Map

Consider the use of a L-nearest neighbor classifier. For a particular sample x, if most of its neighbors share the same class label as x, then the classification of x will be easy. Otherwise, the classification of x will be unreliable or even incorrect. In the following, let $\exp\{-\|x_i - x_j\|^2/\delta^2\}$ be the similarity between samples x_i and x_j. We can thus define the *uncertainty* u_i of a training sample x_i as:

$$u_i = \frac{\sum_{j\in N_i^L, j\notin \pi_{c_i}} \exp\{-\|x_i - x_j\|^2/\delta^2\}}{\sum_{j\in N_i^L} \exp\{-\|x_i - x_j\|^2/\delta^2\}}, \quad (1)$$

where N_i^L is the set of L-nearest neighbors[1] of x_i. From the definition, a large u_i means that the neighboring samples are likely to be of different classes, and hence the classification of x_i is more uncertain. On the contrary, a small u_i indicates that more neighboring samples share the same class label of x_i.

Note that computing the uncertainty relies not only on the data distribution, but also on the label information. Moreover, intuitively, the uncertainty will be high for those training samples lying close to the decision boundary.

2.2. Responsibility Mixture Model

In this section, we propose the *responsibility mixture model* (RMM) for modeling the uncertainty distribution of the data. The RMM is a mixture of K Gaussians[2], with each mixture component normally distributed as $N(m^k, \Sigma^k)$ with mean m^k and covariance matrix Σ^k. The (combined) responsibility distribution function $r(x|\theta)$ at a particular sample x is then

$$r(x|\theta) \equiv \sum_{k=1}^{K} w^k \frac{1}{\sqrt{|\Sigma^k|}\pi^{\frac{m}{2}}} \times$$

[1] In the experiments, we simply use $L = n_c$ for the cth class.

[2] Here, K is chosen empirically.

$$\exp\{-\frac{1}{2}(x-m^k)^T(\Sigma^k)^{-1}(x-m^k)\},$$

where w^k is the prior probability of the kth component (with $\sum_k w^k = 1$). Here, we use $\theta = \{w^k, m^k, \Sigma^k\}_{k=1}^K$ to denote all the model parameters.

Obviously, the local classifiers should be placed near the decision boundary, where classification is the most difficult. As mentioned in Section 2.1, by construction, the uncertainty is also high at those training samples lying close to the decision boundary. Consequently, the mixture should have a high responsibility for areas with high uncertainties. In other words, $r(x_i|\theta)$ should be large when u_i is large, and vice verse.

To achieve this goal, we maximize the following objective function

$$F(u, X \mid \theta) = \sum_{i=1}^{N} u_i \log r(x_i|\theta). \qquad (2)$$

Notice that when the label information is not available, we cannot compute the uncertainty using Eq.(1) and all the u_i's may be taken as one. In this case, Eq.(2) reduces to the log-likelihood

$$\log P(X \mid \theta) = \log \left[\prod_{i=1}^{N} p(x_i \mid \theta)\right].$$

of the standard Gaussian mixture (GMM) that considers the data distribution of the samples only.

Another advantage of this formulation is that areas of high uncertainty should have training samples from at least two classes. Thus, when the local classifiers are placed in those areas, they will not suffer from the above-mentioned problem that only one class of training samples are available in that local region.

2.3. EM for Parameter Estimation

It is difficult to optimize Eq.(2) directly. Therefore, we treat it as an incomplete data problem and use the Expectation Maximization (EM) algorithm (Bilmes, 1998).

Let $z_i^k = 1$ when x_i is generated from the kth mixture component, and 0 otherwise. Concatenating all these missing data together, we have $z_i = [z_i^1, z_i^2, \ldots, z_i^K]^T$, and $Z = \{z_i^k \mid i = 1, \ldots, N; k = 1, \ldots, K\}$. We can then rewrite the objective function in Eq.(2) with the complete data as

$$F(Z, u, X \mid \theta) = \sum_{i=1}^{N} u_i \log p(x_i, z_i \mid \theta), \qquad (3)$$

where

$$p(x_i, z_i \mid \theta) = \sum_{k=1}^{K} z_i^k w^k \frac{1}{\sqrt{\mid \Sigma^k \mid} \pi^{\frac{m}{2}}} \times \exp\{-\frac{1}{2}(x_i - m^k)^T(\Sigma^k)^{-1}(x_i - m^k)\}.$$

Since only one z_i^k is non-zero for a given i, Eq.(3) can be rewritten as

$$F(Z, u, X \mid \theta) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_i^k u_i \log\{w^k \frac{1}{\sqrt{\mid \Sigma^k \mid} \pi^{\frac{m}{2}}} \times \exp[-\frac{1}{2}(x_i - m^k)^T(\Sigma^k)^{-1}(x_i - m^k)]\}.$$

Let $\theta_n = \{w_n^k, m_n^k, \Sigma_n^k\}_{k=1}^K$ be the parameter estimated from the nth step. The Q-function (Bilmes, 1998) is then obtained as

$$\begin{aligned} Q(\theta \mid \theta_n) &= E_z\{F(Z, u, X \mid \theta) \mid X, \theta_n\} \qquad (4) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} E_z(z_i^k) u_i \log\{w^k p(x_i \mid m^k, \Sigma^k)\}, \end{aligned}$$

where the expectation is computed using $p(Z|X, \theta_n)$.

2.3.1. E-Step

From the Bayes rule,

$$E_z(z_i^k) = \frac{w_n^k p(x_i \mid m_n^k, \Sigma_n^k)}{\sum_{j=1}^{K} w_n^j p(x_i \mid m_n^j, \Sigma_n^j)}. \qquad (5)$$

Notice that only the $E_z(z_i^k)$ term is related to Z in Eq.(4). Once we obtain $E_z(z_i^k)$ from Eq.(5), Eq.(4) is only related to the parameter $\theta = \{w^k, m^k, \Sigma^k\}_{k=1}^K$.

2.3.2. M-Step

From Eq.(4), we have

$$\begin{aligned} \frac{\partial Q(\theta \mid \theta_n)}{\partial m^k} &= \sum_{i=1}^{N} E_z(z_i^k) u_i \frac{\partial}{\partial m^k} \log p(x_i \mid m^k, \Sigma^k) \\ &= \sum_{i=1}^{N} E_z(z_i^k) u_i (\Sigma^k)^{-1}(x_i - m^k). \end{aligned}$$

On setting $\partial Q(\theta \mid \theta_n)/\partial m^k$ to zero,

$$m^k = \sum_{i=1}^{N} E_z(z_i^k) u_i x_i / \sum_{i=1}^{N} E_z(z_i^k) u_i,$$

as Σ^k is non-singular. Similarly, on setting $\partial Q(\theta \mid \theta_n)/\partial \Sigma^k = 0$, we have

$$\Sigma^k = \left[\sum_{i=1}^{N} \frac{E_z(z_i^k) u_i}{\sum\limits_{i=1}^{N} E_z(z_i^k) u_i} (x_i - m^k)(x_i - m^k)^T\right]^{-1}.$$

For the prior probability of each mixture component w_k, we use the method of Lagrange multipliers to enforce the constraint $\sum_k w^k = 1$. The Lagrangian is:

$$f(\lambda, w^k) = Q(\theta \mid \theta_n) + \lambda \left(\sum_k w^k - 1 \right).$$

Setting $\partial f(\lambda, w^k)/\partial w^k = 0$, we have

$$\lambda w^k = \sum_{i=1}^{N} E_z(z_i^k) u_i,$$

and therefore

$$w^k = \sum_{i=1}^{N} E_z(z_i^k) u_i / \sum_{k=1}^{K} \sum_{i=1}^{N} E_z(z_i^k) u_i.$$

We can then optimize the parameters of the responsibility mixture model iteratively until convergence.

2.4. Discussion

The GMM is unsupervised and all the samples are considered equally important. If there is prior information on the importance u_i of sample x_i, we can rewrite the objective function of GMM as

$$P(X \mid \theta) = \prod_{i=1}^{N} [p(x_i \mid \theta)]^{u_i}.$$

If u_i is set to be the uncertainty of sample x_i, we can immediately recover the RMM. Thus, while the standard GMM considers only the data distribution, the RMM can be viewed as a way of assigning importance to the training samples based on the label information. An illustration of the superiority of RMM over GMM is shown in Figure 2.

3. Ensemble of Locally Adaptive Classifiers

Locally adaptive classifiers are usually superior to the use of a single global classifier. In this section, we use an ensemble of local non-parametric classifiers for locally adaptive classification based on the class responsibilities obtained from the responsibility mixture model in Section 2.2.

3.1. Training

Using the responsibility mixture model in Section 2.2, the weight of each local classifier for sample x_i is the probability $E_z(z_i^k)$ that x_i belongs to the kth cluster. We now perform local dimensionality reduction by associating a transformation matrix A^k with each local classifier. As discussed in Section 2.1, a particular sample x will be more likely to be classified correctly if most of its neighbors share the same class label as x. We may then define the probability that sample x_i will be correctly classified by the kth local model as

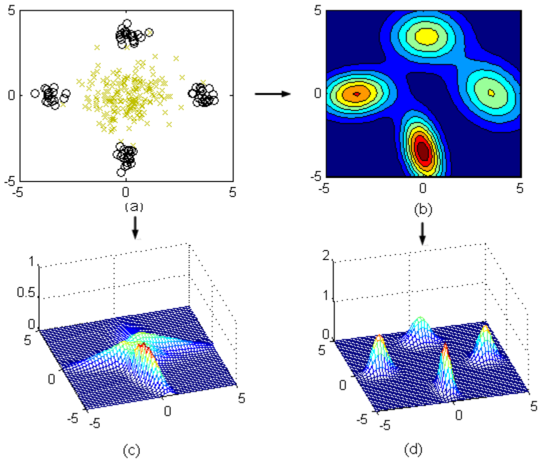


Figure 2. RMM better characterizes the locations of the locally adaptive classifiers. (a) Toy data; (b) Uncertainty map; (c) GMM result; (d) RMM result.

$$p_i^k = \sum_{j:c_j=c_i} \frac{\exp\{-\|A^k x_i - A^k x_j\|^2\}}{\sum_{o \neq i} \exp\{-\|A^k x_i - A^k x_o\|^2\}}. \tag{6}$$

Notice that Eq.(6) is related to the definition of the uncertainty in Eq.(1). In particular, when the number of nearest neighbors L is set to $N-1$ and $A^k = I$, we have $p_i^k + u_i = 1$. For computational efficiency, this simplified formulation will always be used.

Recall that sample x_i belongs to the kth cluster with probability $E_z(z_i^k)$, and that x_i will be correctly classified by the kth local model with probability p_i^k. Thus, to find the optimal $A = \{A^k\}_{k=1}^K$, we maximize

$$G(A) = \sum_{i=1}^{N} \sum_{k=1}^{K} E_z(z_i^k) p_i^k. \tag{7}$$

Denote $x_{ij} = x_i - x_j$. The gradient vector of $G(A)$ can be easily obtained as

$$\begin{aligned} \frac{\partial G(A)}{\partial A} &= \left[\frac{\partial G(A)}{\partial A^1}, \frac{\partial G(A)}{\partial A^2}, \ldots, \frac{\partial G(A)}{\partial A^K} \right]', \quad (8) \\ \frac{\partial G(A)}{\partial A^k} &= -2A^k \sum_{i=1}^{N} \sum_{j:c_j=c_i} E_z(z_i^k) \times \\ & \qquad (p_{ij}^k x_{ij} x_{ij}^T - p_{ij}^k \sum_{o \neq i} p_{io}^k x_{io} x_{io}^T), \end{aligned}$$

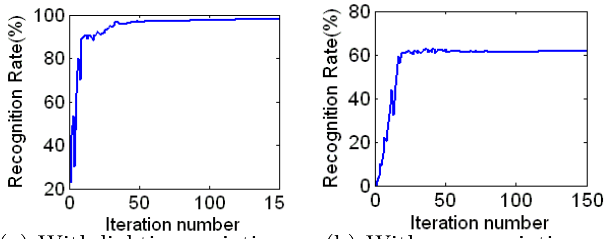
where $p_{ij}^k = \frac{\exp\{-\|A^k x_i - A^k x_j\|^2\}}{\sum_{o \neq i} \exp\{-\|A^k x_i - A^k x_o\|^2\}}$.

However, $G(A)$ may be non-convex and have local optima. In the following, we use simulated annealing (Casella & Robert, 1999) to perform the optimization. The whole algorithm is shown in Algorithm 1. To accelerate optimization, we update the transformation matrix A with a random gradient matrix. In practice, this greatly alleviates the computational complexity without sacrificing accuracy.

Algorithm 1 Simulated Annealing for Parameter Optimization.

```
1: Initialize the transformation matrix $A = A(0)$, and set the iteration counter $n = 0$.
2: Compute the gradient matrix $\frac{\partial G(A(n))}{\partial A(n)}$ using Eq.(8).
3: Compute $B = A(n) - \alpha \frac{\partial G(A(n))}{\partial A(n)} + \varepsilon$, where $\alpha$ is the step size, and $\varepsilon$ is a random matrix of the same size as $A$.
4: Compute $G(B), G(A(n))$ and set $dG = G(B) - G(A(n))$.
5: Set $A(n+1) = B$ with probability $\min(\exp(dG/T_n), 1)$, where $T_n$ is the temperature at the $n$th iteration; otherwise, set $A(n+1) = A(n)$.
6: Set $T_{n+1} = \beta T_n$ (where $\beta < 1$ is the cooling rate), $n = n + 1$, then, go to step 2.
```

As an illustration, Figure 3 demonstrates the improvement in classification accuracy with the number of iterations. The two experiments are performed on a subset of the CMU PIE database (Sim et al., 2003) with 8,442 facial images of 67 subjects[3]. The algorithm is initialized by setting the local transformation matrix A^k to be the identity matrix, the temperature to 200, and $\beta = 0.925$. As can be seen, the proposed algorithm converges in only about 50 iterations.



(a) With lighting variation: (b) With pose variations.

Figure 3. Recognition accuracies (%) vs. the number of iterations for the proposed method.

3.2. Testing

Given a new sample x, we first compute the responsibilities of the various local models as

$$z^k(x) = \frac{w^k p(x \mid \theta^k)}{\sum_{j=1}^{K} w^j p(x \mid \theta^j)}.$$

Its class label can then be predicted according to

$$\arg\max_i \sum_{k=1}^{K} z^k(x) \frac{\exp\{-\|A^k x - A^k x_i\|^2\}}{\sum_{j=1}^{N} \exp\{-\|A^k x - A^k x_j\|^2\}}.$$

3.3. Discussion

If the number of clusters K is set to one, then the objective function in Eq.(7) reduces to

$$G(A) = \sum_{i=1}^{N} \sum_{c_j = c_i} \frac{\exp\{-\|Ax_i - Ax_j\|^2\}}{\sum_{o \neq i} \exp\{-\|Ax_i - Ax_o\|^2\}}.$$

This is the same as that of *neighborhood component analysis* (NCA) (Goldberger et al., 2004), which is a linear dimensionality reduction method for learning a Mahalanobis distance for nearest neighbor classifiers. Therefore, NCA is a special case of the RMM with only one cluster. Besides, NCA uses gradient descent to compute the solution, and may easily fall into a local optimum. Moreover, NCA is computationally expensive, especially when a large number of training samples is available.

4. Experiments

In this section, we perform a number of experiments on both toy and real-world data sets. The proposed method will be referred to as *Locally Adaptive Classification Piloted by Uncertainty* (LCU).

4.1. Dimensionality Reduction

To visualize the effectiveness of RMM, we perform dimensionality reduction by combining RMM with the Fisher criterion (FDA) in a manner similar to that in (Kim & Kittler, 1994). Experiments are performed on two sets of artificial data (Figures 4(a) and (b)), where the task is to reduce the data from 2-D to 1-D. Comparisons are made with three traditional dimensionality reduction algorithms: PCA, LDA and locally linear discriminant analysis (LLDA) (Kim & Kittler, 1994). For both LLDA and LCU, the number of clusters K is set to 4 for the first data set and 3 for the second one.

Results are shown in Figures 4(c)-(j). As can be seen, the two classes become more separated by using the

[3] Please refer to Section 4.2 for the detailed experimental setups.

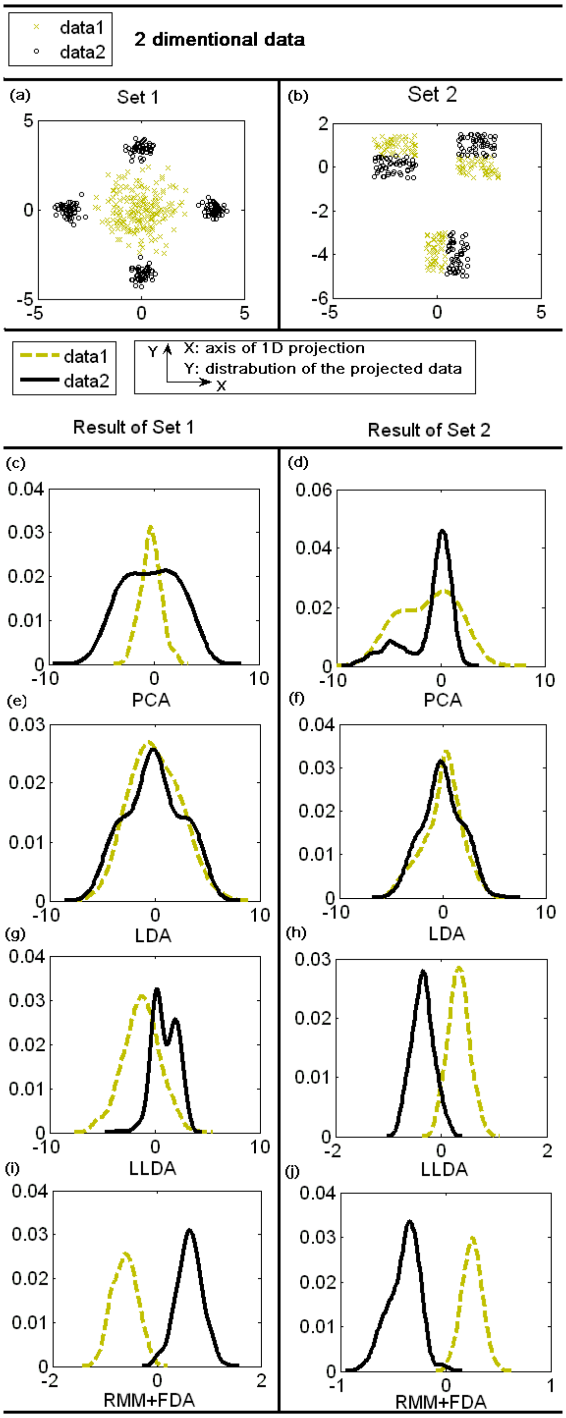


Figure 4. Dimensionality reduction results. (a) Data set 1; (b) Data set 2; (c)-(j): Distributions of the 1-D projections of the two data sets obtained using various methods ((c),(d) PCA; (e),(f) LDA; (g),(h) LLDA; (i),(j) LCU).

proposed method. As discussed earlier, this is due to the facts that both PCA and LDA are globally linear, and LLDA is based directly on the data distribution; while RMM can make use of the uncertainty of the data (see also Figure 2).

4.2. Face Recognition

In this section, we perform face recognition experiments on the CMU PIE face database (Sim et al., 2003) (Figure 5). We use a total of 8,442 images from 67 subjects[4] that were acquired under different lighting conditions and poses. As pre-processing, PCA is applied to these images and the number of principal components extracted is determined by keeping about 95% of the total energy.

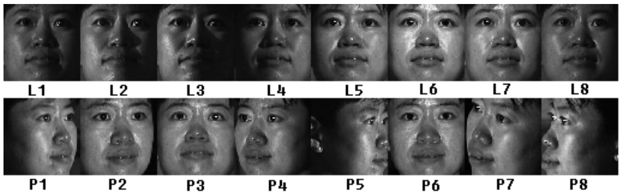


Figure 5. Examples of cropped images from the CMU PIE face database (with different lighting conditions and poses).

LCU is compared to three popular face recognition schemes: Eigenface (Turk & Pentland, 1991), Fisherface (Belhumeur et al., 1997) and LLDA. For Fisherface and LLDA, we try all possible numbers of extracted features (from 1 to 66) and report the best recognition results; whereas for LCU, we simply fix the number of features to $N_c - 1$. As there are 7 different lighting conditions in the training set, we set the number of clusters/components K to 7 for both LCU and LLDA. The transformation matrices of Eigenface (PCA), Fisherface (LDA), LLDA and LCU are shown in Figure 6.

4.2.1. Varying the Lighting

We first study the algorithms' robustness to lighting variations. The training set consists of 2,345 images from 67 subjects, with 5 poses and 7 different lighting conditions. The test set consists of 2,680 images from the same 67 subjects, with the same poses but 8 different lighting conditions. As can be seen from Table 1, LCU performs much better than the use of a single model. This indicates the superiority of the fusing strategy in LCU. Also, the results show that LCU outperforms LLDA, which is a locally adaptive algorithm based on the data distribution.

[4] One subject does not have the complete set of images for all lighting conditions and poses, and so is not used.

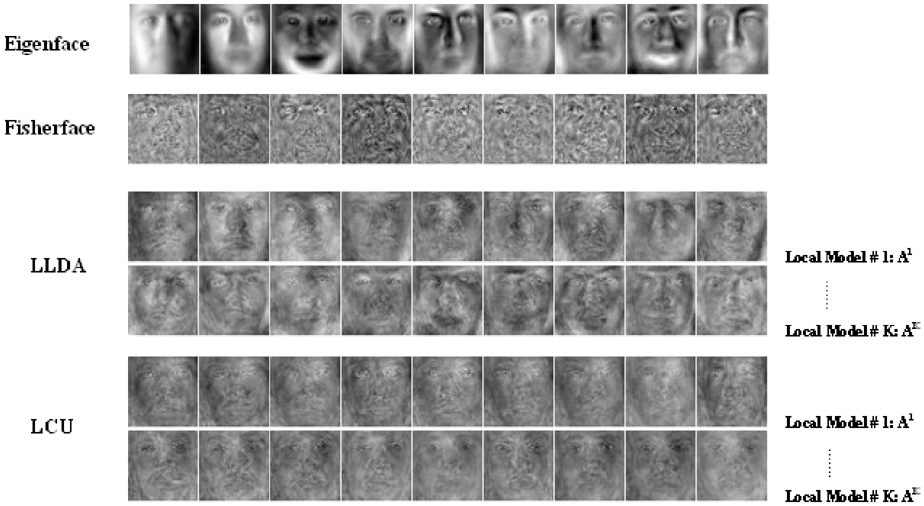


Figure 6. Visualization of the column vectors of the transformation matrices obtained by Eigenface (PCA), Fisherface (LDA), LLDA and LCU on the training set of the CMU PIE database.

Table 1. Recognition accuracies (%) on the CMU PIE database with different lighting conditions.

light	*Eigenface*	*Fisherface*	*LLDA*	*LCU*
L1	48.76	79.37	95.05	**97.75**
L2	51.10	80.65	94.29	**96.06**
L3	44.05	80.20	93.84	**97.87**
L4	45.77	77.43	93.85	**98.48**
L5	53.06	87.74	99.10	**99.50**
L6	58.88	89.29	99.53	**99.65**
L7	56.21	88.17	98.63	**99.73**
L8	52.74	85.03	98.17	**99.02**
average	51.32	83.46	96.56	**98.51**

Table 2. Recognition accuracies (%) on the CMU PIE database with different poses.

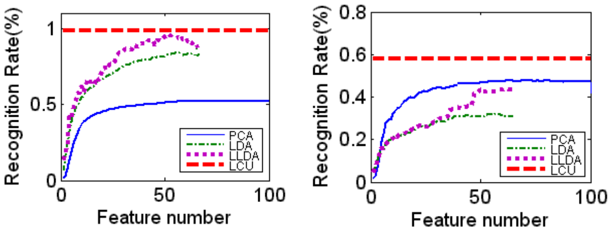
pose	*Eigenface*	*FisherFace*	*LLDA*	*LCU*
P1	32.67	20.12	29.87	**46.73**
P2	60.53	47.76	62.47	**71.49**
P3	61.41	46.59	60.09	**68.75**
P4	36.12	24.31	31.65	**56.63**
P5	37.70	26.79	27.74	**54.54**
P6	64.91	39.11	63.38	**69.83**
P7	48.83	25.55	40.26	**57.74**
P8	44.07	22.91	35.81	**48.92**
average	48.28	31.64	43.91	**59.33**

4.2.2. Varying the Pose

In this experiment, we study the robustness to pose variations. We use face images with 5 poses and 7 lighting conditions for training, and images with 8 unseen poses but the same lighting variations for testing. Table 2 shows the recognition results. Eigenface and Fisherface are sensitive to pose variations and perform much worse than in Section 4.2.1. Note that Eigenface outperforms Fisherface here. A similar observation is also made in (Martinez & Kak, 2001). Again, LCU is the best among all the algorithms evaluated here.

4.2.3. Number of Features Extracted

As shown in Figure 7, the performance of Eigenface, Fisherface and LLDA all depend on the number of features extracted. On the other hand, for LCU, if the sizes of its local transform matrices (A^k's) are large enough, unnecessary dimensions can be automatically removed in the optimization process. Hence, there is no need to test different numbers of extracted features. As mentioned earlier, we have simply fixed it at $N_c - 1$ in all the experiments.



(a) With lighting variations. (b) With pose variations.

Figure 7. Recognition accuracies (%) vs. the number of features extracted. Note that the number of features for LCU is always fixed at $N_c - 1$.

5. Conclusion and Future Works

In this paper, we introduce the responsibility mixture model and an ensemble of classifiers with local dimensionality reduction for locally adaptive classification. The responsibility mixture model ensures that the local classifiers are placed near the potential decision boundary, where they will be most effective for classification purposes. It also avoids the situation where only one class of samples is available for training the local classifier. Parameters of the ensemble are then learned by directly optimizing an estimate of the probability that the samples will be correctly classified with a nearest neighbor classifier.

In the future, we will study algorithms similar to the loc-boost (Meir et al., 2000) that can combine a set of localized adaptive classifiers guided by the uncertainty map. Moreover, the support vector machine (Osuna et al., 1997) and many boosting/leveraging algorithms (Meir et al., 2000) also consider the decision boundary for discriminant analysis. While most of these were originally proposed for binary classification, LCU is naturally suitable for multi-class classification. The underlying relationship between LCU and these algorithms is an interesting direction for further research.

References

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 711–720.

Bilmes, J. A. (1998). *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models* (Technical Report). International Computer Science Institute, Berkeley.

Bregler, C., & Omohundro, S. (1995). Nonlinear image interpolation using manifold learning. *Advances in Neural Information Processing Systems* (pp. 973–980).

Casella, G., & Robert, C. P. (1999). *Monte Carlo statistical methods*. Springer-Verlag.

Chapelle, O., Weston, J., Bottou, L., & Vapnik, V. (2000). Vicinal risk minimization. *Neural Information Processing Systems* (pp. 416–422).

Fisher, R. A. (1936). The use of multiple measures in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

Ghahramani, Z., & Hinton, G. (1996). *The EM algorithm for mixtures of factor analyzers* (Technical Report CRG-TR-96-1). University of Toronto.

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdino, R. (2004). Neighborhood component analysis. *Neural Information Processing Systems* (pp. 513–520).

Kim, T.-K., & Kittler, J. (1994). Locally linear discriminant analysis for multimidally distributed classed for face recognition with a single model image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 318–327.

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 228–233.

Meir, R., El-Yaniv, R., & Ben-David, S. (2000). Localized boosting. *Proceeding of The 13th Annual Conference on Computational Learning Theory*, 190–199.

Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machine: An application to face detection. *Proceedings of the International Conference on Computer Vision and Pattern recognition* (pp. 130–136).

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Selim, S. Z., & Ismail, M. A. (1984). k-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 81–87.

Sim, T., Baker, S., & Bsat, M. (2003). The CMU pose, illumination, and expression database. *Proceeding of the European Conference on Computer Vision* (pp. 1615–1618).

Toussaint, M., & Vijayakumar, S. (2005). Learning discontinuities with products-of-sigmoids for switching between local models. *Proceedings of the 22nd international conference on Machine Learning* (pp. 904–911).

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *13*, 71–86.

Yan, S., Zhang, H., Hu, Y., Zhang, B., & Cheng, Q. (2004). Discriminant analysis on embedded manifold. *Proceeding of the European Conference on Computer Vision* (pp. 121–132).