FCM Classifier for High-dimensional Data

Hidetomo Ichihashi, *Member, IEEE,* Katsuhiro Honda, *Member, IEEE,* Akira Notsu, and Eri Miyamoto

***Abstract*— A fuzzy classifier based on the fuzzy c-means (FCM) clustering has shown a decisive generalization ability in classification. The FCM classifier uses covariance structures to represent flexible shapes of clusters. Despite its effectiveness, the intense computation of covariance matrices is an impediment for classifying a set of high-dimensional data. This paper proposes a way of directly handling high-dimensional data in the FCM clustering and classification. The proposed classifier without any preprocessing outperforms the k-nearest neighbor (k-NN) classifier with PCA on the benchmark set of COREL image collection.**

I. Introduction

THIS paper proposes a way of directly handling high-dimensional data set in the fuzzy c-means (FCM) clustering [1] by which covariance structures of the clusters are taken into account. The FCM clustering is applied to a classifier design and the particle swarm optimization (PSO) [2], [3] is introduced for parameter optimization.

The Gaussian mixture models (GMM) or normal mixture [4] can be seen as a clustering method, though it is a method of representing a data distribution by a weighted sum of the density functions of normal populations. The classifier based on the GMM clustering is called a neural network [5] or the Gaussian mixture classifier (GMC). The classifier based on the FCM clustering is called the fuzzy c-means classifier (FCMC) [6], [7], [8]. For the detail derivation of the algorithm and the classification performance, see [9](WCCI'08).

In the FCMC, the standard FCM clustering objective function [1] is slightly generalized and the iteratively reweighted least square (IRLS) technique [10] is applied [11]. Cluster memberships are defined by a function of Mahalanobis distances between data vectors and cluster centers.

We adopt a post-supervised design, in which the unsupervised clustering is done on a per class basis in the first phase of FCMC. It is implemented by using the data from one class at a time. When working with the data class by class, the prototypes that are found for each labeled class already have the assigned physical labels. The sum of the cluster membership functions plays the role of the discriminant function for classification.

High performance classifiers usually have parameters to be selected. For example, the support vector machine (SVM) [12], [13] has the margin and kernel parameters. These parameter values are selected by some optimization procedures to improve the generalization ability.

For the parameter optimization of our classifier, we apply the PSO algorithm in the second phase of FCMC. PSO is an evolutionary algorithm and is easy-to-implement. PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as the genetic algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In the past several years, PSO has been successfully applied in many research and application areas.

The authors are with the Department of Computer Science and Intelligent Systems, Osaka Prefecture University, 1-1 Gakuen-cho, Nakaku, Sakai, Osaka 599-8531 Japan (e-mail: {ichi, honda, notsu}@cs.osakafu-u.ac.jp).

FCMC uses covariance structures to represent flexible shapes of clusters. Despite its effectiveness in the accuracy of classification [8], the intense computation of covariance matrices and their eigenvalue decomposition is an impediment for classifying a set of high-dimensional data. Instead of covariance matrices, we utilize matrices in terms of inner products of data vectors. The FCM clustering and classification based on Mahalanobis distances are carried out from the inner product matrices. The proposed approach can handle high-dimensional data without any preprocessing for feature extraction. Let the number of data vectors be N and the number of their dimensions be p, then the computation of an inner product matrix is $O(N^2p)$, whereas that of a covariance matrix is $O(Np^2)$. Let X be an $N \times p$ matrix of mean corrected row vectors (i.e., the set of centered data). Then $X^\top X/N$ is the covariance matrix, which is used in FCMC. $XX^\top$ is the matrix in terms of inner product. The proposed classifier is the dual of the FCMC and produces the same clustering and classification results with that of FCMC. If the dimensionality of features is larger than the number of samples, it is more efficient than FCMC.

The paper is organized as follows. Section II gives a brief description of the generalized FCM clustering and the classifier design based on the FCM clustering. Parameter optimization with PSO is proposed. An FCM clustering by using matrices in terms of inner products instead of covariance matrices is detailed in Section III. Section IV provides the results of numerical experiments. The subset of COREL image database of James Wang (http://wang.ist.psu.edu/docs/related/) [14], [15] is used for performance evaluation. The subset consists of 1,000 pictures, each of which is a 300,000 dimensional datum. Section V concludes the paper.

II. FCM Classifier

In this section, we summarize the two phases of FCMC. See [9](WCCI'08) for more detail. The first phase is the unsupervised clustering.

Squared Mahalanobis distance from $x_k \in \mathcal{R}^p$ to $v_i \in \mathcal{R}^p$

978-1-4244-1819-0/08/$25.00©2008 IEEE

is written as:

$$D(x_k, v_i; S_i) = (x_k - v_i)^\top S_i^{-1}(x_k - v_i) \quad (1)$$

where S_i is a covariance matrix of vectors x_k in the i-th cluster.

$$S_i = \frac{\sum_{k=1}^{N} u_{ki}(x_k - v_i)(x_k - v_i)^\top}{\sum_{k=1}^{N} u_{ki}}. \quad (2)$$

$$v_i = \frac{\sum_{k=1}^{N} u_{ki} x_k}{\sum_{k=1}^{N} u_{ki}} \quad (3)$$

is the cluster center and the mixing proportion of i-th cluster is

$$\alpha_i = \frac{\sum_{k=1}^{N} u_{ki}}{\sum_{j=1}^{c}\sum_{k=1}^{N} u_{jk}} = \frac{1}{n}\sum_{k=1}^{N} u_{ki}. \quad (4)$$

We minimize the IRLS-FCM objective function:

$$J_{ifcm} = \sum_{i=1}^{c}\sum_{k=1}^{N} w(d_{ki})\left(D(x_k, v_i; S_i) + \log|S_i|\right), \quad (5)$$

where the adaptive weight $w(d_{ki})$ is replaced by the membership function as $w(d_{ki}) = u_{ki}$. Covariance matrix S_i in (2) and cluster center v_i can be derived by differentiating (5) with respect to S_i and v_i respectively.

The weight w should be recomputed after each iteration in order to be used in the next iteration. In the robust M-estimation [10], [16], the function $w(d_{ki})$ provides an adaptive weight. The influence from x_k is decreased when $|x_k - v_i|$ is very large and suppressed when it is infinitely large. While IRLS approach in general does not guarantee the convergence to a global minimum, experimental results have shown reasonable convergence points.

To facilitate competitive movements of cluster centers, we need to define the weight function normalized as:

$$u_{ki} = \frac{u^*_{ki}}{\sum_{l=1}^{c} u^*_{lk}}. \quad (6)$$

u^* is defined as:

$$u^*_{ki} = \frac{\alpha_i |S_i|^{-\frac{1}{\gamma}}}{(D(x_k, v_i; S_i)/0.1 + \nu)^{\frac{1}{m}}}. \quad (7)$$

u_{ki} of (6) can be rewritten as:

$$u_{ki} = \alpha_i |S_i|^{-\frac{1}{\gamma}} \times \left[\sum_{j=1}^{c}\left(\frac{D(x_k, v_i; S_i)/0.1 + \nu}{D(x_k, v_j; S_i)/0.1 + \nu}\right)^{\frac{1}{m}} \alpha_j |S_j|^{-\frac{1}{\gamma}}\right]^{-1}. \quad (8)$$

u^* is a modified and parameterized multivariational version of Cauchy's weight function in the M-estimator or of the probability density function (PDF) of Cauchy distribution.

In the post-supervised classifier design, the clustering is implemented by using the data from one class at a time. The prototypes (cluster centers) that are found for each labeled class already have the assigned physical labels.

After completing the clustering for all classes, the classification is performed by computing class memberships. Let π_q denote the mixing proportion of class q, i.e., the $a\ priori$ probability of class q. The class membership of k-th data x_k to class q is computed as:

$$u^*_{qjk} = \frac{\alpha_{qj} |S_{qj}|^{\frac{1}{\gamma}}}{(D_q(x_k, v_j; S_j)/0.1 + \nu)^{\frac{1}{m}}}, \quad (9)$$

$$\tilde{u}_{qk} = \frac{\pi_q \sum_{j=1}^{c} u^*_{qjk}}{\sum_{s=1}^{Q} \pi_s \sum_{j=1}^{c} u^*_{sjk}}, \quad (10)$$

where c denotes the number of clusters of each class and Q denotes the number of classes. The denominator in (10) can be disregarded when applied solely for classification. Whereas (8) is referred to as a classification function for clustering, (10) is a discriminant function for pattern classification. The FCM classifier performs somewhat better than alternative approaches [8], [9] and requires comparable computation time with GMC because the functional structure of FCM is similar to that of GMM.

The modification of covariance matrices in the mixture of probabilistic principal component analysis (MPCA) [17] or the character recognition [18] is applied to the FCM classifier. P_i is a $p \times p$ matrix of eigenvectors of S_i. p equals the dimensionality of input samples. Let S'_i denote an approximation of S_i in (2). P^r_i is a $p \times r$ matrix of eigenvectors corresponding to the r largest eigenvalues, where $r < p-1$ for the approximation of S_i. Δ^r_i is an $r \times r$ diagonal matrix of δ_{il}, $l = 1, ..., r$, i.e., the square root of the eigenvalues.

$$S'_i = P^r_i((\Delta^r_i)^2 - \sigma_i^2 I_r)P^{r\top}_i + P_i(\sigma_i^2 I_p)P_i^\top. \quad (11)$$

Inverse of S'_i becomes

$$S_i^{'-1} = P^r_i((\Delta^r_i)^{-2} - \sigma_i^{-2} I_r)P^{r\top}_i + \sigma_i^{-2} I_r, \quad (12)$$

$$\sigma_i^2 = (\mathrm{trace}(S_i) - \Sigma_{l=1}^{r}\delta_{il}^2)/(p-r). \quad (13)$$

When r=0, S'_i is reduced to a unit matrix and $D(x_k, v_i; S_i)$ in (1) is reduced to Euclidean distance.

The FCM classifier has several parameters, whose best values are not known in advance, consequently some kinds of model selection (parameter search) must be done. The goal is to identify good values so that the classifier can accurately predict unseen data (i.e., testing/checking data). Because it may not be useful to achieve high training accuracy (i.e., accuracy on training data whose class labels are known), a common way is to separate training data to two parts of which one is considered unknown in training the classifier. The prediction accuracy on this set can then more precisely reflect the performance on classifying unknown data.

In our proposed approach, parameters m, ν and γ of the membership function are optimized in the post-supervised classification phase by using PSO. In PSO, the potential solutions are called particles. The particles fly through the problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far. The solution is evaluated by the fitness value, which is also stored. This value is called "pbest". Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the swarm. The best value is a global best and is called "gbest". The search for the better positions follows the rule as:

$$\begin{aligned} Para^{t+1} &= Para^t + Velo^{t+1} && (14)\\ Velo^{t+1} &= w_0\ Velo^t + c_1\ Rand_1\ (pbest - Para^t) \\ &+ c_2\ Rand_2\ (gbest - Para^t), && (15)\end{aligned}$$

where $Para$ is the parameter vector (e.g., m, γ, ν of FCMC) to be optimized and $Velo$ is the vector of their velocity. It should be noted that $Rand$ is a diagonal matrix of random numbers chosen from the unit interval [0, 1]. w_0, c_1 and c_2 are scalar constants. $pbest$ and $gbest$ are the vectors of positions of pbest and gbest respectively. The rule by (15) is the standard PSO, though written in vector-matrix form. The best setting of the parameters (i.e., m, γ, ν) is picked to minimize the error rate on test sets.

III. FCM Classifier for High-dimensional Data

In order to alleviate the intense computation of covariance matrices and their eigenvalue decomposition, we utilize matrices in terms of inner products of data vectors. The FCM clustering by using the matrices is detailed in this section. The fuzzy covariance matrix for the i-th cluster is written as:

$$\begin{aligned} S_i &= \frac{1}{N\alpha_i} X_i^\top M_i X_i \\ &= ((N\alpha_i)^{-\frac{1}{2}} M_i^{\frac{1}{2}} X_i)^\top ((N\alpha_i)^{-\frac{1}{2}} M_i^{\frac{1}{2}} X_i), \quad (16)\end{aligned}$$

where $X_i = (x_1 - v_i, ..., x_N - v_i)^\top$ and $M_i = \mathrm{diag}(u_1, ..., u_N)$ is the diagonal matrix whose diagonal elements are $(u_1, ..., u_N)$.

Eigenvalue decomposition of S_i is written as:

$$S_i = P_i \Delta_i^2 P_i^\top, \quad (17)$$

where $P_i^r = (p_{i1}, ..., p_{ir})$ is a $p \times r$ matrix and $p_{i1}, ..., p_{ir} \in \mathcal{R}^p$ are the eigenvectors or PCA basis vectors associated with the positive eigenvalues $(\delta_{i1}^2, ..., \delta_{ir}^2)$ of S_i. Vectors are normalized as $p_{il}^\top p_{il} = 1$. $\Delta_i^2 = \mathrm{diag}(\delta_{i1}^2, ..., \delta_{ir}^2)$ is the diagonal matrix of the eigenvalues. Through singular value decomposition

$$(N\alpha_i)^{-\frac{1}{2}} M_i^{\frac{1}{2}} X_i = F_i \Delta_i P_i^{r\top}, \quad (18)$$

we have

$$X_i P_i^r \Delta_i^{-1} = (N\alpha_i)^{\frac{1}{2}} M_i^{-\frac{1}{2}} F_i, \quad (19)$$

where F_i is an $N \times r$ matrix.

If S_i is invertible and $\mathrm{rank}(S_i) = r = p$, Mahalanobis distance between x_k and v_i is written from (19) as:

$$\begin{aligned} D(x_k, v_i; S_i) &= (x_k - v_i)^\top S_i^{-1} (x_k - v_i) \\ &= (x_k - v_i)^\top P_i^r \Delta_i^{-2} P_i^{r\top} (x_k - v_i) \\ &= ((x_k - v_i)^\top P_i^r \Delta_i^{-1}) \\ &\quad \times ((x_k - v_i)^\top P_i^r \Delta_i^{-1})^\top \\ &= N\alpha_i u_{ki}^{-1} f_{ki}^\top f_{ki}, && (20)\end{aligned}$$

where $F_i = (f_{i1}, ..., f_{iN})^\top$.

We define $N \times N$ matrix K_i to obtain F_i and Δ_i as:

$$K_i = (N\alpha_i)^{-1} M_i^{\frac{1}{2}} X_i X_i^\top M_i^{\frac{1}{2}}. \quad (21)$$

Let $X_0 = (x_1, ..., x_N)^\top$ be a data matrix, then

$$X_i = (I_N - \mathbf{1}_N \overline{u_i}^\top) X_0, \quad (22)$$

where I_N is a unit matrix of dimension N. $\mathbf{1}_N$ is the vector of dimension $N \times 1$ with all entries equal to 1.

$$\overline{u_i} = (u_{i1} / \sum_{k=1}^N u_{ki}, ..., u_{iN} / \sum_{k=1}^N u_{ki})^\top. \quad (23)$$

$X_i X_i^\top$ can be written as:

$$\begin{aligned} X_i X_i^\top &= (I_N - \mathbf{1}_N \overline{u_i}^\top) X_0 X_0^\top (I_N - \overline{u_i} \mathbf{1}_N^\top) \\ &= X_0 X_0^\top - X_0 X_0^\top \overline{u_i} \mathbf{1}_N^\top \\ &\quad - \mathbf{1}_N \overline{u_i}^\top X_0 X_0^\top \\ &\quad + \mathbf{1}_N \overline{u_i}^\top X_0 X_0^\top \overline{u_i} \mathbf{1}_N^\top. && (24)\end{aligned}$$

By using (18), K_i is rewritten as:

$$\begin{aligned} K_i &= ((N\alpha_i)^{-\frac{1}{2}} M_i^{\frac{1}{2}} X_i)((N\alpha_i)^{-\frac{1}{2}} M_i^{\frac{1}{2}} X_i)^\top \\ &= (F_i \Delta_i P_i^{r\top})(P_i^r \Delta_i F_i^\top) \\ &= F_i \Delta_i^2 F_i^\top. && (25)\end{aligned}$$

So, F_i and Δ_i are obtained from eigenvalue decomposition of K_i. If $\mathrm{rank}(S_i) = r = p$, the value of $|S_i|$ required for updating u_i is:

$$|S_i| = |P_i^r||\Delta_i^2||P_i^{r\top}| = |\Delta_i^2| = \prod_{l=1}^p \delta_{il}^2 \quad (26)$$

When the dimensionality is high, the distance by (1) is very large and becomes really difficult to compute. The problem is known as the curse of dimensionality. One measure taken to reduce the dimensionality is to use small number of PCA basis vectors. When the number of samples is small and p is large, S_i often becomes singular and noninvertible. We again use the approximation method of S_i in [17], [18]. We pick r' such that $r' < r < p$ then we can write

$$S_i' = P_i^{r'} ((\Delta_i^{r'})^2 - \sigma_i^2 I_{r'}) P_i^{r'\top} + P_i^r (\sigma_i^2 I_r) P_i^{r\top}, \quad (27)$$

though we do not calculate S_i' explicitly since p is large. $I_{r'}$ is a unit matrix of dimension r'. $(\Delta_i^{r'})^2$ is a diagonal matrix whose diagonal elements (δ_{il}^2) are the r' largest

eigenvalues of K_i, $P_i^{r'}$ is a matrix of $p \times r'$ consisted of the r' eigenvectors.

$$\begin{aligned}\sigma_i^2 &= \frac{1}{r-r'}\sum_{l=r'+1}^{r}\delta_{il}^2 \\ &= \frac{1}{r-r'}(\mathrm{trace}(K_i)-\sum_{l=1}^{r'}\delta_{il}^2) \end{aligned} \quad (28)$$

Thus,

$$\begin{aligned} & D(x_k, v_i; S_i') \\ = \; & (x_k-v_i)^\top S_i'^{-1}(x_k-v_i) \\ = \; & \sum_{l=1}^{r'}\frac{1}{\delta_{il}^2}(x_k-v_i)^\top p_{il}\times \\ & (x_k-v_i)^\top p_{il} \\ + \; & \frac{1}{\sigma_i^2}\sum_{l=r'+1}^{r}(x_k-v_i)^\top p_{il}\times \\ & (x_k-v_i)^\top p_{il} \\ = \; & \sum_{l=1}^{r'}\frac{1}{\delta_{il}^2}(x_k-v_i)^\top p_{il}\times \\ & (x_k-v_i)^\top p_{il} \\ - \; & \frac{1}{\sigma_i^2}\sum_{l=1}^{r'}(x_k-v_i)^\top p_{il}\times \\ & (x_k-v_i)^\top p_{il} \\ + \; & \frac{1}{\sigma_i^2}(x_k-v_i)^\top(x_k-v_i) \\ = \; & N\alpha_i u_{ki}^{-1}\left(\sum_{l=1}^{r'}f_{ikl}^2-\frac{1}{\sigma_i^2}\sum_{l=1}^{r'}f_{ikl}^2\delta_{il}^2\right) \\ + \; & \sigma_i^{-2}(x_k-v_i)^\top(x_k-v_i) \\ = \; & N\alpha_i u_{ki}^{-1}f_{ki}'^\top f_{ki}' \\ - \; & \sigma_i^{-2}N\alpha_i u_{ki}^{-1}f_{ki}'^\top(\Delta_i^r)^2 f_{ki}' \\ + \; & \sigma_i^{-2}(x_k-v_i)^\top(x_k-v_i) \end{aligned} \quad (29)$$

where $f_{ki}' = (f_{ik1}, ..., f_{ikr'})^\top$. $(x_k-v_i)^\top(x_k-v_i)$ is the k-th diagonal element of $X_i X_i^\top$.

$$|S_i'| \simeq (\prod_{l=1}^{r'}\delta_{il}^2)\sigma_i^{2(r-r')} \quad (30)$$

It should be noted that r need not be precisely equal to the number of positive eigenvalues. It should be smaller than that so that the determinants $|S_i'|$ are all positive. Since small positive eigenvalues can be disregarded, r is relatively small. When $r-r'$ is small such as 20 or 40, the classifier's accuracy is usually better than that with larger ones.

The algorithm has the following steps.

Algorithm FCMCH: FCMC for High-dimensional Data

step1. Partition the dataset of each class into two clusters by PCA scores, i.e., one for positive scores and the other for negative ones. Let v_i be the mean of the i-th cluster. Let the lower limit of m^* be LL and the upper limit be UL. Let $m^* = LL$. Divide the dataset into two folds, i.e., the training set and test set.

step2. Partition the training set by FCM with $m = m^*, \gamma = \nu = 1$. Clustering is on a per class basis. All S_i and v_i are then fixed.

step3. Search for the best m, γ, and ν by PSO, which minimize the classification error rate on the test set.

step4. $m^* := m^* + \Delta$ (Δ: step size). If $m^* > UL$ then terminate, else go to step2.

From (18), we have

$$(N\alpha_i)^{-\frac{1}{2}}X_i^\top M_i^{\frac{1}{2}}F_i\Delta_i^{-2} = P_i^r\Delta_i^{-1}. \quad (31)$$

Let X_0^{NEW} be the matrix of non-centered test data. By multiplying centered test set X_i^{NEW} from the left to both sides of (31), we have

$$\begin{aligned} X_i^{NEW}P_i^r\Delta_i^{-1} &= (N\alpha_i)^{-\frac{1}{2}}X_i^{NEW}X_i^\top M_i^{\frac{1}{2}}F_i\Delta_i^{-2} \\ &= (N\alpha_i)^{-\frac{1}{2}}(X_0^{NEW}X_0^\top \\ &+ X_iX_i^\top - X_0X_0^\top)M_i^{\frac{1}{2}}F_i\Delta_i^{-2}, \end{aligned} \quad (32)$$

which corresponds to the left side of (19). From (3),

$$1_N^\top M_i X_i = 0_p^\top, \quad (33)$$

where 0_p is a zero vector of dimension p. By multiplying $1_N^\top M_i$ to (19) from the left, we have

$$1_N^\top M_i^{\frac{1}{2}}F_i = 0_r^\top, \quad (34)$$

where 0_r is a zero vector of dimension r.

And, for all $l \in \{1, ..., N\}$

$$\begin{aligned} & (x_{k'}-v_i)^\top X_i^\top - x_{k'}^\top X_0^\top \\ - \; & (x_l-v_i)^\top X_i^\top + x_l^\top X_0^\top \\ = \; & (x_{k'}-x_l)^\top X_i^\top - (x_{k'}-x_l)^\top X_0^\top \\ = \; & (x_{k'}-x_l)^\top (X_i-X_0)^\top \\ = \; & -(x_{k'}-x_l)^\top v_i 1_n^\top. \end{aligned} \quad (35)$$

By multiplying $M_i^{\frac{1}{2}}F_i$ to both sides of (35) from the right, and by multiplying X_i^{NEW} to both sides of (31) from the left, we have (32).

Namely, when $x_{k'}$ is given, we have for all $l \in \{1, ..., N\}$,

$$\begin{aligned} & (x_{k'}-v_i)^\top P_i^r\Delta_i^{-1} = (n\alpha_i)^{-\frac{1}{2}}(x_{k'}^\top X_0^\top \\ & +(x_l-v_i)^\top X_i^\top - x_l^\top X_0^\top)M_i^{\frac{1}{2}}F_i\Delta_i^{-2}, \end{aligned} \quad (36)$$

where $(x_{k'}-v_i)$ is the k-th row of X_i^{NEW}. By using (24), (36) can be computed from the matrix in terms of inner product without computing v_i explicitly.

In (29), to compute $D(x_k', v_i; S_i)$, we need Euclidian distance between the new data $x_{k'}$ and the cluster center v_i. Since

$$x_{k'} - v_i = x_{k'} - X_0^\top \overline{u_i}, \quad (37)$$

the squared distance can be written as:

$$||x_{k'} - v_i||^2 = x_{k'}^\top x_{k'} - 2\overline{u_i}^\top X_0 x_{k'} + \overline{u_i}^\top X_0 X_0^\top \overline{u_i}. \quad (38)$$

We have so far explained the procedure when the classifier is trained by the algorithm FCMCH. After the training is completed, we use the trained classifier for unknown new data. Since we can compute v_i from (3), $P_i^{r'}$ is obtained from F_i, Δ_i and the data matrix X_i as:

$$P_i^{r'} = (N\alpha_i)^{-\frac{1}{2}} X_i^\top M_i^{\frac{1}{2}} F_i \Delta_i^{-1}. \quad (39)$$

Therefore we can easily compute the Mahalanobis distances by the second row of (20).

IV. Numerical Experiment

Automatic linguistic indexing of pictures is important to content-based image retrieval and computer object recognition. It can potentially be applied to many areas, including biomedicine, commerce, the military, education, digital libraries, and Web searching.

We used the subset of COREL image database (http:// wang.ist.psu.edu/ docs/related/), which was used by James Wang for tests of his SIMPLIcity System [14] and the statistical modeling approach to automatic linguistic indexing of pictures [15].

Li and Wang [15] trained their system for automatic linguistic indexing of pictures. Instead of annotating the images, the program was used to select the category with the highest likelihood for each test image. That is, they used the classification power of the system as an indication of the annotation accuracy. An image is considered to be annotated correctly if the computer predicts the true category the image belongs to. Although these image categories do not share annotation words, they may be semantically related.

Figs. 1-4 show the examples of the subset images, which is classified into ten categories such as beach and flowers, each of which consists of 100 pictures. Total number of pictures is 1,000 and each image has $256 \times 384 \times 3 \approx$300,000 dimensions. The problem is to correctly classify these images into the 10 categories shown in Table I. We compared the classification performance of FCMCH to that of the k-NN classifier with a preprocessing of PCA. The 1,000 images are divided into two sets, i.e., the training set (667) and the test set (333). The images from each class are evenly included in the training set and the test set.

As shown in Table II, the number of clusters (c) is two for each class. r and r' are chosen by trial and error. Parameters for PSO are also chosen by trial and error, and are shown in Table III. The optimization performance is not so sensitive to these parameters. See [19](WCCI'08), for the comparison in the application of FCMCH to a fMRI study. Table IV shows the parameter values, which minimizes the classification error rate on the test set. For a system which classifies randomly, the average error rate is 90% for the 10-class problem, though the rate by FCMCH is 22.22%.

TABLE I
Classes of COREL Images

ID	class
1	African people and village
2	Beach
3	Building
4	Buses
5	Dinosaurs
6	Elephants
7	Flowers
8	Horses
9	Mountains and glaciers
10	Food

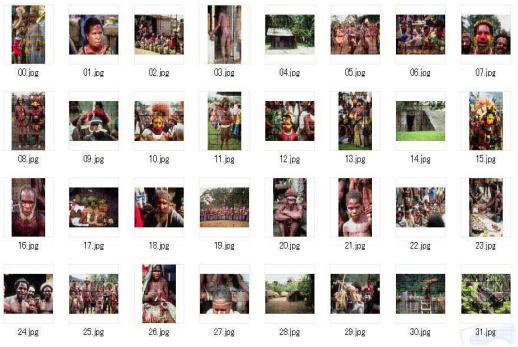

Fig. 1. African people and village

We compared FCMCH to k-NN. The nearest neighbor classifier does not abstract the data, but rather uses all training data to label unseen data objects with the same label as the nearest object in the training set. The nearest neighbor classifier easily overfits to the training data. Accordingly, instead of 1-nearest neighbor, generally k nearest neighboring data objects are considered in the k-NN classifier. The class label of unseen objects is then established by a majority vote. For the parameter of k-NN (i.e.,k), we tested all integer values from 1 to 50. The lowest error rate is 0.4685 when k=6. Table V shows the result by k-NN in which we used PCA as the preprocessing of classification. The lowest error rate is 0.3724 when the number of PC basis vectors is 10 and k =5. The error rate of FCMCH is about 10% smaller than that of k-NN. The per class classification error rate by FCMCH is shown in Fig. 5.

Classification accuracy on the same subset of COREL image database by Li and Wang's approach is 0.636 (Table 3 in [15]), so the error rate is 0.364. The error rate by FCMCH is smaller than that of their approach, though the basis of comparison is not completely the same.

V. Conclusion

We have proposed a fuzzy classifier for high dimensional data, which is based on the FCM clustering and parameter optimization by PSO. The use of inner product matrices alleviates the intense computation of covariance matrices and their eigenvalue decompositions. The proposed FCMCH surpassed the k-NN classifier with PCA preprocessing, though

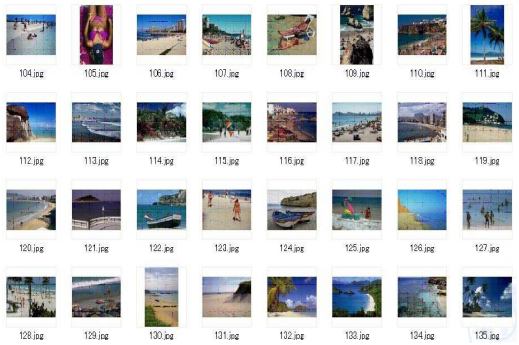

Fig. 2. Beach

Fig. 3. Flowers

FCMCH also uses PCA basis vectors to approximate covariance matrices.

REFERENCES

[1] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.

[2] R.C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory, "*Proc. of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.

[3] J. Kennedy and R.C. Engelbrech, "Particle swarm optimization. , "*Proc of the IEEE International Conference on Neural Networks*, Piscataway, NJ, vol. 4, pp. 1942-1948, 1995.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[5] R. L. Streit and T. E. Luginbuhl, "Maximum likelihood training of probabilistic neural networks," *IEEE Transactions on Neural Networks*, vol.5, no.5, pp.764-783, 1994.

[6] H. Ichihashi and K. Honda, "Fuzzy *c*-means classifier for incomplete data sets with outliers and missing values," *Proc. of the International Conference on Computational Intelligence for Modelling, Control and Automation*, Vienna, November, pp.457-464, 2005.

[7] H. Ichihashi, K. Honda, T. Hattori, "Regularized discriminant in the setting of fuzzy *c*-means classifier," *Proc. of the IEEE World Congress on Computational Intelligence*, Vancouver, Canada, pp.4266-4271, 2006.

[8] H. Ichihashi, K. Honda, A. Notsu and T. Yagi, "Fuzzy *c*-means classifier with deterministic initialization and missing value imputation," *Proc. of the 2007 IEEE Symposium on Foundations of Computational Intelligence*, Hawaii, April, pp. 214-221, 2007.

[9] H. Ichihashi, K. Honda, A. Notsu and K. Ohta, "Fuzzy *c*-means classifier with particle swarm optimization," *Proc. of the IEEE World Congress on Computational Intelligence*, Hong Kong, China, June 1-6, 2006.

Fig. 4. Food

TABLE II

PARAMETER VALUES OF FCMCH

c	m^*	r'	r
2	0.9	15	45

[10] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares, "*Communications in Statistics*, vol. A6, no. 9, pp. 813-827, 1977.

[11] H. Ichihashi, K. Honda, A. Notsu and T. Hattori, "Aggregation of standard and entropy based fuzzy c-means clustering by a modified objective function," *Proc. of the 2007 IEEE Symposium on Foundations of Computational Intelligence*, Hawaii, April, pp. 447-453, 2007.

[12] V.N. Vapnik, *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, Berlin, 1982.

[13] C. Cortes and V. Vapnik, "Support-vector network." *Machine Learning*, vol.20, pp.273-297, 1995.

[14] J. Z. Wang, J. Li and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries, "*IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 974-963, 2001.

[15] J. Li and J. Z. Wang, " Automatic linguistic indexing of pictures by a statistical modeling approach, "*IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.25, no. 9, pp. 1075-1088, 2003.

[16] P. J. Huber. *Robust Statistics*. New York:Wiley, first edition, 1981.

[17] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analysers, "*Neural Computation*, vol.11, pp.443-482, 1999.

[18] F. Sun, S.Omachi, and H. Aso, "Precise selection of candidates for hand written character recognition," *IEICE Trans. Information and Systems*, vol.E79-D, no.3, pp.510-515, 1996

[19] H. Ichihashi, K. Honda, A. Notsu and T. Hattori, "Classifier of BOLD signals from active and inactive brain states using FCM clustering and evolutionary algorithms," *Proc. of the IEEE World Congress on Computational Intelligence*, Hong Kong, China, June 1-6, 2006.

TABLE III

PARAMETER VALUES OF PSO

number of particles	10
number of iterations	50
w_0	0.9
c_1	0.4
c_2	0.4

TABLE IV

PARAMETER VALUES CHOSEN BY PSO AND ERROR RATE

m	γ	ν	classification error rate
0.2667	14.3252	173.8158	0.2222

TABLE V

CLASSIFICATION ERROR RATES OF k-NN WITH PCA PREPROCESSING

number of PC vectors	k	error rate
500	1	0.7538
50	5	0.4565
20	1	0.4144
12	13	0.3964
11	21	0.3784
10	5	0.3724
9	11	0.4114
6	31	0.4324
5	13	0.4144
4	12	0.4414
1	30	0.6426

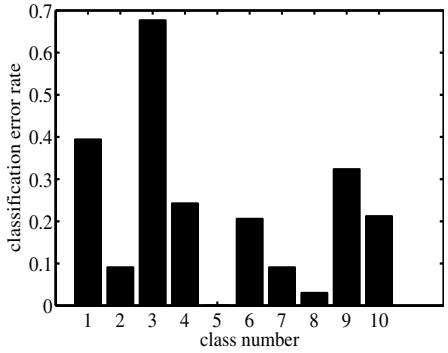

Fig. 5. Per class error rate by FCMCH on the subset of COREL image database