PAPER

Pattern Recognition with Gaussian Mixture Models of Marginal Distributions

Masako OMACHI[†], *Member and* **Shinichiro OMACHI**[††a)], *Senior Member*

SUMMARY Precise estimation of data distribution with a small number of sample patterns is an important and challenging problem in the field of statistical pattern recognition. In this paper, we propose a novel method for estimating multimodal data distribution based on the Gaussian mixture model. In the proposed method, multiple random vectors are generated after classifying the elements of the feature vector into subsets so that there is no correlation between any pair of subsets. The Gaussian mixture model for each subset is then constructed independently. As a result, the constructed model is represented as the product of the Gaussian mixture models of marginal distributions. To make the classification of the elements effective, a graph cut technique is used for rearranging the elements of the feature vectors to gather elements with a high correlation into the same subset. The proposed method is applied to a character recognition problem that requires high-dimensional feature vectors. Experiments with a public handwritten digit database show that the proposed method improves the accuracy of classification. In addition, the effect of classifying the elements of the feature vectors is shown by visualizing the distribution.

key words: *pattern recognition, Gaussian mixture model, graph cut, small sample size problem, character recognition*

1. Introduction

In many statistical pattern recognition methods, it is important to precisely estimate data distribution. In general, precise estimation of the distribution requires a great number of sample patterns, especially when the feature vector obtained from the pattern is high dimensional. However, for some pattern recognition problems, such as face recognition or character recognition, very high-dimensional feature vectors are necessary, and there are not always enough sample patterns for estimating the distributions. Precisely estimating the distribution with a limited number of sample patterns is still a challenging problem.

The Gaussian mixture model is a linear combination of Gaussian probability density functions, and it is used for representing multimodal distributions. It is used not only for pattern recognition [1], [2] but also in many fields such as prediction [3], [4], monitoring [5], [6], segmentation [7], [8], discrimination [9], and clustering [10], [11]. The parameters of the Gaussian mixture model can be iteratively estimated by calculating the probability of each sample pattern.

In this paper, we propose a novel method for estimating data distribution with the help of the Gaussian mixture model. As described in detail in Sect. 2.2, when calculating the distribution of patterns, it may be effective in some cases to classify the elements of feature vectors into subsets and calculate the distribution for each subset independently. This strategy is identical to block diagonalization of the covariance matrix by substituting zeros for the non-block-diagonal elements, and it has been used to reduce computational time for solving pattern recognition problems [12], [13]. In these methods, the information of the non-block-diagonal elements is lost by block diagonalization. Researchers have investigated rearranging the elements to decrease the loss. Koshiba et al. minimized the difference between the distances calculated with the original and the block-diagonalized covariance matrices for each possible combination of elements [12], which was computationally very expensive for high-dimensional feature vectors. Sun et al. proposed an ad-hoc method for iteratively changing the elements of the feature vector; however, it offered no guarantee of convergence [13].

In the proposed method, to avoid losing the correlation information, multiple random vectors are generated by variable transformation so that there is no correlation between any pair of vectors after classifying the elements. The Gaussian mixture model is then independently constructed for each subset so that no information is lost. As a result, the constructed model can be represented as a product of the Gaussian mixture models of marginal distributions. To classify the elements of the feature vector, a graph cut technique [14] is used to gather elements with a high correlation into the same subset.

The proposed method is general with no limit on target applications. To show the effect of the proposed method, it is applied to a character recognition problem that requires high-dimensional feature vectors. Experiments are carried out with a public handwritten digit database MNIST [15]. The experimental results demonstrate that the proposed method improves the classification accuracy and is more effective when the sample size is small. In addition, the effect of classifying the elements of the feature vectors is shown by visualizing the distribution.

2. Proposed Method

The key idea of the proposed method is to classify the elements of the feature vectors into subsets and generate a set of new vectors corresponding to the subsets. The Gaussian

Manuscript received February 16, 2010.
Manuscript revised September 16, 2010.

[†]The author is with the Advanced Course of Production System and Design Engineering, Sendai National College of Technology, Natori-shi, 981–1239 Japan.
[††]The author is with the Graduate School of Engineering, Tohoku University, Sendai-shi, 980–8579 Japan.
a) E-mail: machi@ecei.tohoku.ac.jp

DOI: 10.1587/transinf.E94.D.317

mixture model corresponding to each subset is then constructed. To make the classification of the elements effective, a graph cut technique is used for rearranging the elements of the feature vectors to gather the elements with a high correlation into the same subset.

The flowchart of the proposed method is displayed in Fig. 1. Given the sample feature vectors, first the elements of the feature vectors are rearranged by a graph cut approach. They are then classified into subsets and new random vectors are generated with the rearranged vectors. Finally, the Gaussian mixture model is constructed for each subset.

2.1 Gaussian Mixture Model

Gaussian mixture models are used for approximating multimodal distributions. A Gaussian mixture model of vector $\boldsymbol{x}$ is a linear combination of n Gaussian probability density functions and is defined as

$$g(\boldsymbol{x}) = \sum_{k=1}^{n} p_k f(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k). \tag{1}$$

Here,

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{(d(\boldsymbol{x})/2)}|\Sigma|^{1/2}} \times \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\} \tag{2}$$

is a component density function, p_k is a mixing parameter that satisfies $\sum_{k=1}^{n} p_k = 1$, $d(\boldsymbol{x})$ is the dimensionality of $\boldsymbol{x}$, $\boldsymbol{\mu}$ is a $d(\boldsymbol{x})$-dimensional mean vector and Σ is a $d(\boldsymbol{x}) \times d(\boldsymbol{x})$ covariance matrix.

Given the sample vectors, the parameters of the model can be estimated by the expectation-maximization algorithm [16]. The probability of each component density function is calculated for each sample vector, and the parameters of the component density function are iteratively updated according to the probability. Given m sample vectors $\{\boldsymbol{x}_j\}$ and a set of initial parameters $\{p_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \Sigma_k^{(0)}\}$, the parameters are estimated by applying the following formulae.

$$p_k^{(t+1)} = \frac{1}{m}\sum_{j=1}^{m} \tilde{p}_k^{(t)}(\boldsymbol{x}_j), \tag{3}$$

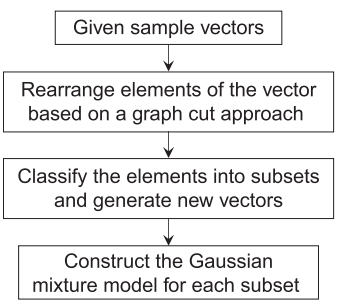


Fig. 1 Flow of the proposed method.

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{j=1}^{m} \tilde{p}_k^{(t)}(\boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=1}^{m} \tilde{p}_k^{(t)}(\boldsymbol{x}_j)}, \tag{4}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{j=1}^{m} \tilde{p}_k^{(t)}(\boldsymbol{x}_j)(\boldsymbol{x}_j-\boldsymbol{\mu}_k^{(t)})(\boldsymbol{x}_j-\boldsymbol{\mu}_k^{(t)})^T}{\sum_{j=1}^{m} \tilde{p}_k^{(t)}(\boldsymbol{x}_j)}, \tag{5}$$

where

$$\tilde{p}_k^{(t)}(\boldsymbol{x}_j) = \frac{p_k^{(t)} f(\boldsymbol{x}_j; \boldsymbol{\mu}_k^{(t)}, \Sigma_k^{(t)})}{\sum_{l=1}^{n} p_l^{(t)} f(\boldsymbol{x}_j; \boldsymbol{\mu}_l^{(t)}, \Sigma_l^{(t)})}. \tag{6}$$

Equations (3), (4), (5) and (6) are calculated iteratively until convergence.

2.2 Classifying the Elements of the Feature Vector

The probability for each feature vector is calculated according to Eq. (6). However, when estimating a Gaussian mixture model, it may be effective in some cases to classify the elements of the feature vectors into subsets and independently calculate the distribution for each subset. An intuitive example is shown in Fig. 2. These are some of the handwritten digit images included in the MNIST database [15]. Figures 2 (a) and (b) show images having similar upper parts and dissimilar lower parts. On the other hand, the images in Figs. 2 (b) and (c) have similar lower parts and dissimilar upper parts. In this case, it would be effective to divide the images into upper and lower parts. Then, for the upper part, Figs. 2 (a) and (b) can be used for estimating one component density function, and Figs. 2 (c) and (d) can be used for estimating another component density function. On the other hand, for the lower part, Figs. 2 (a) and (d) can be used for estimating one component density function, and Figs. 2 (b) and (c) can be used for estimating another component density function.

For simplicity, suppose that the elements of each feature vector are classified into two subsets. Given a d-dimensional vector, we can rearrange the elements of the vector and construct a new d-dimensional vector $\boldsymbol{x}$ so that

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} \tag{7}$$

is satisfied, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ include only the elements of the first and second subsets, respectively. Let the corresponding mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ be

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \tag{8}$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}. \tag{9}$$

(a) (b) (c) (d)

Fig. 2 Example of handwritten digit images.

The strategy of classifying the elements into subsets and independently calculating the distributions is equivalent to independently calculating the distribution of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. In this case, Σ_{12} and Σ_{21} are regarded as a zero matrix, and Eq. (9) becomes a block-diagonal matrix. In the case of a Gaussian distribution, the probability density function of Eq. (2) can be calculated as the product of two marginal distributions.

$$f(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = f(\boldsymbol{x}_1;\boldsymbol{\mu}_1,\Sigma_{11})f(\boldsymbol{x}_2;\boldsymbol{\mu}_2,\Sigma_{22}). \tag{10}$$

However, it is rare that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are exactly independent, and ignoring the correlation of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ sometimes deteriorates the classification accuracy. To avoid losing the correlation information, in the proposed method, multiple random vectors are created by transforming the variables so that there is no correlation between any pair of vectors [17] after classifying the elements. Let

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix} \tag{11}$$

where

$$\boldsymbol{y}_1 = \boldsymbol{x}_1 - \Sigma_{12}\Sigma_{22}^{-1}\boldsymbol{x}_2, \tag{12}$$

$$\boldsymbol{y}_2 = \boldsymbol{x}_2. \tag{13}$$

Then the mean vector $\boldsymbol{\mu}_y$ and the covariance matrix Σ_y will be

$$\boldsymbol{\mu}_y = \begin{pmatrix} \boldsymbol{\mu}_1 - \Sigma_{12}\Sigma_{22}^{-1}\boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \tag{14}$$

$$\Sigma_y = \begin{pmatrix} \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} & \boldsymbol{O} \\ \boldsymbol{O} & \Sigma_{22} \end{pmatrix}. \tag{15}$$

Note that the original vector can be represented by any number of vectors by repeating this procedure.

The distribution of $\boldsymbol{y}_2$ is the marginal distribution of $\boldsymbol{x}$, and the elements of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ have no correlations. In the proposed method, the Gaussian mixture models of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are independently constructed.

As a by-product, the computational time of the probability density function is reduced. If the dimensionality of $\boldsymbol{x}_1$ is half of that of $\boldsymbol{x}$, the computational time for Eq. (10) will be half of that for Eq. (2). Considering the calculation time for Eq. (12), the total computational time is reduced to three-fourths of that of the original.

2.3 Rearrangement of Elements

We now explain how to rearrange the elements of the original feature vector as per Eq. (7). Since the proposed method independently estimates the distributions of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, it will be more effective if there is a high correlation between the elements of the same vector. In other words, it will be effective to make the values of the elements of Σ_{12} or Σ_{21} as small as possible. For this purpose, we apply a graph cut approach for rearranging the elements of the original feature vector.

Suppose the dimensionality of the feature vector is d and consider a complete graph $G = (V, E)$ that has d nodes $v_i \in V$. Each node corresponds to an element of the feature vector, and the edge weight $w(u, v)$ is the absolute value of the (u, v)th element of the covariance matrix of the feature vector.

Consider partitioning G into two parts. We use the normalized cut (Ncut) [14] as the criterion that should be minimized for graph partition. Suppose A and B are exclusive subsets of V that satisfy $A \cup B = V$. Ncut is defined as

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{16}$$

where

$$cut(A, B) = \sum_{u\in A, v\in B} w(u, v), \tag{17}$$

$$assoc(A, V) = \sum_{u\in A, t\in V} w(u, t). \tag{18}$$

Here, the sets of the elements of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of Eq. (7) correspond to A and B, respectively. The sum of the elements of Σ_{12} or Σ_{21} in Eq. (9) corresponds to $cut(A, B)$. Therefore, minimizing Eq. (16) will diminish the values of the elements of Σ_{12} or Σ_{21}.

Although this minimization is NP-complete, by relaxing it to the real value domain, a reasonable solution can be derived as follows [14]: Let W be the matrix of $w(i, j)$, D be a diagonal matrix with $D_{ii} = \sum_j w(i, j)$, and $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_d\}$ be the indicator vector of the partition.

$$\alpha_i = \begin{cases} 1 & \text{if } v_i \in A \\ -1 & \text{if } v_i \in B \end{cases}. \tag{19}$$

The graph partition can be obtained by solving

$$(D - W)\boldsymbol{\beta} = \lambda D\boldsymbol{\beta}, \tag{20}$$

for the eigenvectors where

$$\boldsymbol{\beta} = (\mathbf{1} + \boldsymbol{\alpha}) - \frac{\sum_{\alpha_i>0} D_{ii}}{\sum_{\alpha_i\le 0} D_{ii}}(\mathbf{1} - \boldsymbol{\alpha}), \tag{21}$$

and $\mathbf{1}$ is a vector in which all the elements are one. This problem can be changed to the following standard eigenvalue problem.

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}\boldsymbol{\gamma} = \lambda\boldsymbol{\gamma}, \tag{22}$$

where $\boldsymbol{\gamma} = D^{\frac{1}{2}}\boldsymbol{\beta}$. Solving Eq. (22) for the eigenvector with the second smallest eigenvalue, the graph can be partitioned into two parts according to the sign of each element. In the proposed method, we rearrange the elements of the feature vector according to the values of the elements of this eigenvector.

2.4 Regularization

In general, when the number of sample vectors is small and the dimensionality of the vector is large, regularization [18] is effective. In the framework of the well-known MQDF (modified quadratic discriminant function) [19],

similar technique is introduced.

For regularization, $\Sigma + \delta I$ is used instead of Σ in Eq. (2), where I is a $d(\boldsymbol{x}) \times d(\boldsymbol{x})$ identity matrix and δ is a positive constant. This strategy may be combined with the proposed method for better results. The proposed method is combined with regularization and the performance is tested in an experiment.

3. Experiment

3.1 Data and Features

We carried out an experiment to confirm the effectiveness of the proposed method. For the experiment, we used a handwritten digit database, MNIST [15]. MNIST includes a training set and a test set of images of single-digit numbers in ten categories (from "0" to "9").

We used the directional element feature [20] as the feature of the character images. An input image is normalized to 64×64 dots, and a contour of the image is extracted. Then, orientation—vertical, horizontal, or slanted at ±45°—is assigned for each pixel. The image is divided into 49 sub-areas of 16×16 dots where each sub-area overlaps eight dots with the adjacent sub-area. For each sub-area, a four-dimensional vector is defined to represent the quantities of the four orientations. The dimensionality of this feature is 196 ($= 4 \times 49$).

Given an unknown character image, the feature vector is extracted from this image. Recognition is achieved by calculating the probability of belonging to each category and finding the category with the maximum probability. We compared the proposed method with the traditional Gaussian mixture model that uses Eq. (1) as it is; hereafter called the traditional method. The number of component density functions was four for both the traditional and the proposed methods. In the proposed method, the feature vector was classified into two subsets with $d/2 (= 98)$ elements.

Note that the numbers of subsets and the component density functions are very important parameters that influence the performance. If necessary, suitable values of the parameters can be found by cross-validation performed by dividing the training patterns to improve the classification accuracy. However, since we would rather focus on the performances of the traditional and the proposed methods, we fixed these parameters in the experiment.

3.2 Initial Parameters

To estimate the parameters of the Gaussian mixture model, we must choose a set of initial parameters. In order to determine these parameters, the mean vector $\hat{\boldsymbol{\mu}}$ and the covariance matrix $\hat{\Sigma}$ are calculated with m sample vectors for each category.

$$\hat{\boldsymbol{\mu}} = \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{x}_j, \tag{23}$$

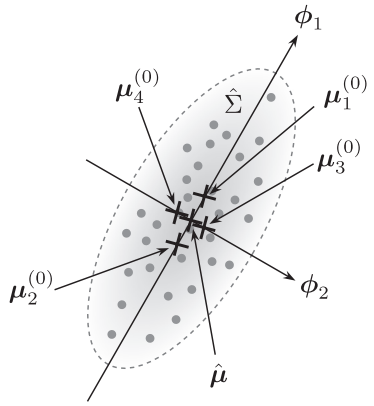


Fig. 3 Initial parameters.

$$\hat{\Sigma} = \frac{1}{m-1} \sum_{j=1}^{m} (\boldsymbol{x}_j - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}})^T. \tag{24}$$

Let λ_i be the ith eigenvalue of $\hat{\Sigma}$ sorted in descending order, and $\boldsymbol{\phi}_i$ be the eigenvector with λ_i. We determine the initial parameters as

$$p_k^{(0)} = \frac{1}{n}, \tag{25}$$

$$\boldsymbol{\mu}_k^{(0)} = \hat{\boldsymbol{\mu}} - (-1)^k \varepsilon \boldsymbol{\phi}_{\lceil k/2 \rceil}, \tag{26}$$

$$\Sigma_k^{(0)} = \hat{\Sigma}, \tag{27}$$

where n is the number of component density functions and ε is a small constant. Figure 3 displays the initial parameters for the case of $n = 4$.

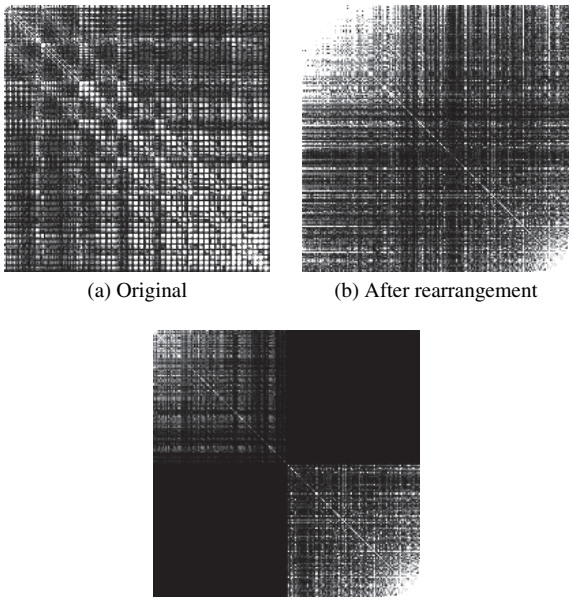
3.3 Example of Covariance Matrix

To show the effect of rearranging the elements of the feature vectors, a representative covariance matrix is displayed in Fig. 4. The absolute value of each element of the covariance matrix is represented by the brightness.

Figure 4 (a) is a 196×196 covariance matrix calculated with the feature vectors obtained from the first 200 images of "7" in the training set. With this covariance matrix, the elements of each feature vector are rearranged by the algorithm described in Sect. 2.3. Figure 4 (b) is the covariance matrix obtained with the rearranged feature vectors. Compared to Fig. 4 (a), brighter pixels are gathered in the block diagonals of Fig. 4 (b). Figure 4 (c) is the block-diagonal covariance matrix of $\boldsymbol{y}$ obtained by the algorithm described in Sect. 2.2.

3.4 Example of Data Distribution

To show the effect of classifying the elements of feature vectors, the data distribution is displayed. Figures 5 (a) and (b) display the distributions of the patterns of "7" for two subsets. For visualization, sample vectors are projected onto the $\phi_1\phi_2$ plane, i.e., the plane defined by the first and second principal components, and plotted as crosses.



(a) Original (b) After rearrangement

(c) After block diagonalization

Fig. 4 Covariance matrices.

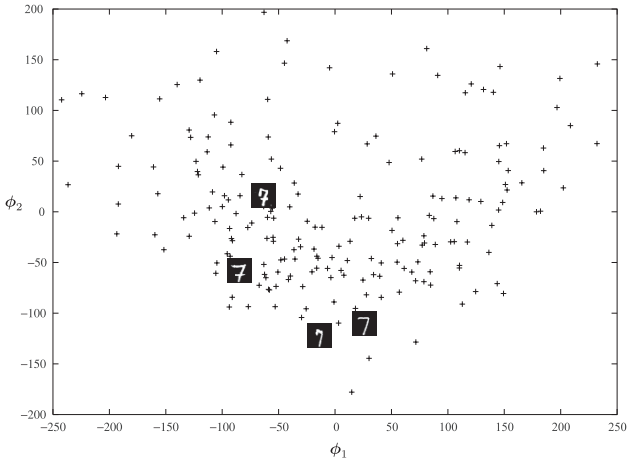
The position of each image of Fig. 2 is also displayed in the figure. The distribution of these four images is quite different in each figure. In Fig. 5 (a), the images that have similar lower parts are closely placed; it seems that most of the elements of the feature vectors that correspond to the lower part of the character image are included in subset 1. On the other hand, the images in which the upper parts are similar are closely placed in Fig. 5 (b).

3.5 Results

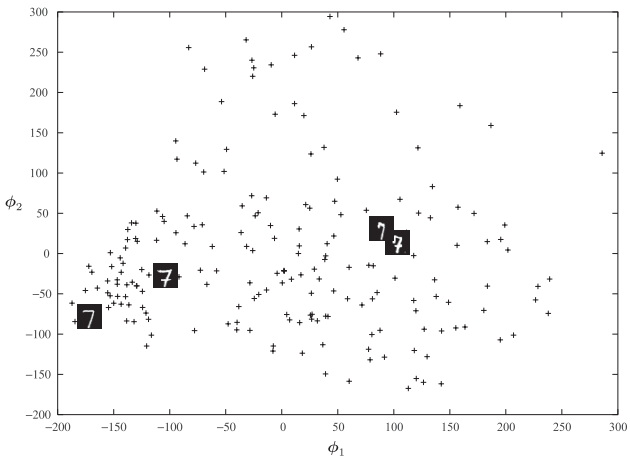
All the character images in the test set were recognized by the models constructed with the sample patterns in the training set. The number of sample patterns for each category was changed from 100 to 600. The sample patterns used for training were chosen randomly from the training set, and this trial was repeated ten times. Figure 6 displays the average and the standard deviation of the accuracy in recognition. In every case, the accuracy of the proposed method was better than that of the traditional method. The results demonstrated that the proposed method is especially effective when the number of sample patterns is small.

To check whether the difference in the average accuracy is statistically significant, the *t*-test was carried out. It was found that the differences were statistically significant with a significance level of 0.01 when the number of sample patterns was less than or equal to 400. If the significance level is 0.05, the difference of 500 sample patterns was also significant.

Table 1 shows the average computational time for each method. The computational time is the time required for calculating the probability for ten categories and selecting the category with the maximum probability. The time for preprocessing (extracting feature vectors from images), which



(a) Subset 1



(b) Subset 2

Fig. 5 Data distribution of patterns of "7".

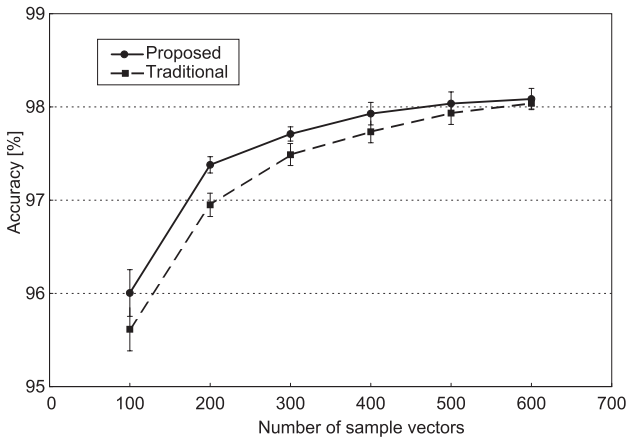


Fig. 6 Average and standard deviation of accuracy in recognition.

Table 1 Average computational time.

Proposed method	Traditional method
6.18 msec	7.96 msec

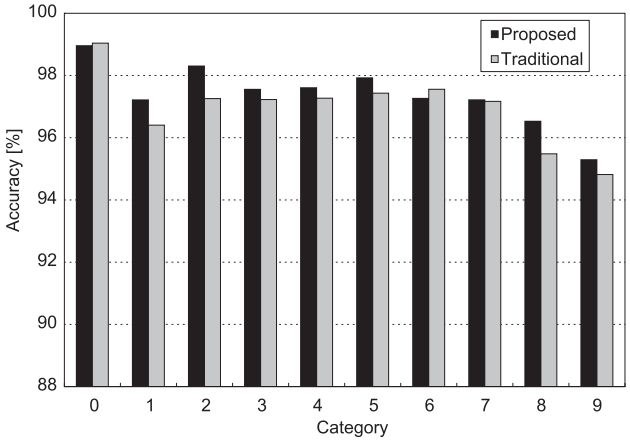


Fig. 7 Average accuracy for each category.

(a) (b) (c) (d)

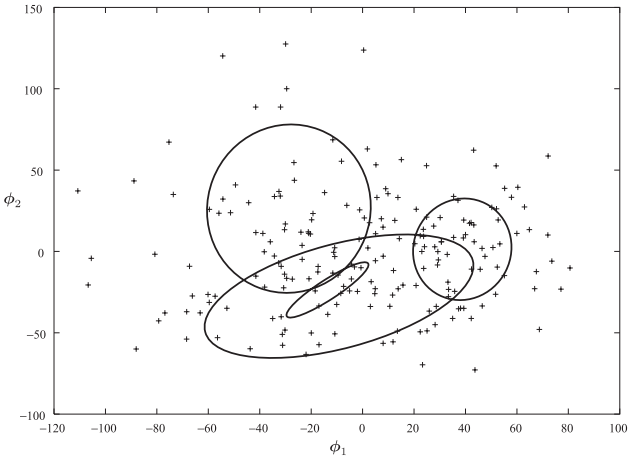
Fig. 8 Example of images correctly recognized by the proposed method.

requires 0.90 milliseconds for each image, is not included. This table shows that the proposed method not only improves the recognition accuracy but also reduces the computational time.

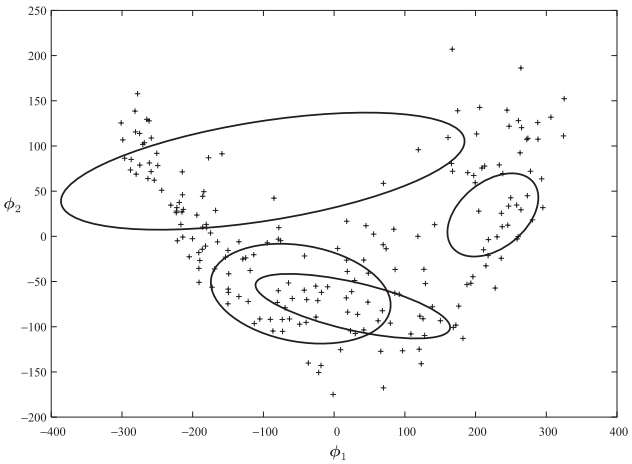
Figure 7 displays the average accuracy for each category when the number of samples is 200. This figure shows that the accuracy was improved for almost all the categories; the difference is noticeably large for categories "1," "2" and "8." For these categories, since the variations in the character patterns are large, more complicated statistical models are necessary. The proposed method could model the variations of the character patterns and achieve a higher accuracy than the traditional method.

Figure 8 displays some examples of the character images that were correctly recognized by the proposed method and were misclassified by the traditional method. Figures 8 (a), (b) and (c) were recognized as "2" and Fig. 8 (d) was recognized as "3" by the traditional method. This figure reveals that category "8" has many variations and that the proposed method possibly modeled these variations more flexibly than the traditional method.

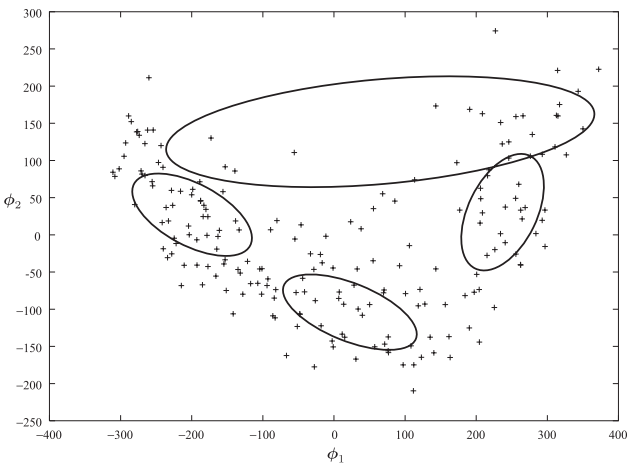
To show the behavior of the proposed method in the case of category "8," the component density functions for each subset constructed by the proposed method are shown in Figs. 9 (a) and (b). For visualization, the sample vectors and the hyperellipsoid of equal probability for each component density function are projected onto the $\phi_1\phi_2$ plane. Figure 9 (c) displays the component density functions obtained by the traditional method and the sample vectors. In this case, the original distribution is very complicated, as shown in Fig. 9 (c). Since the proposed method constructed the Gaussian mixture models after classifying the elements



(a) Proposed method (subset 1)



(b) Proposed method (subset 2)



(c) Traditional method

Fig. 9 Data distribution and component density functions.

of the feature vector into subsets, appropriate models could be constructed for two kinds of distributions.

Although the accuracy was improved for almost all the categories, the accuracy deteriorated for the cases of "0" and "6," as shown in Fig. 7. Figure 10 displays some examples of the character images that were misclassified by

(a) (b) (c)

Fig. 10 Example of images misclassified by the proposed method.

the proposed method and were correctly recognized by the traditional method. All of these images were recognized as "5" by the proposed method. Some images in category "5" are similar to "6." In the proposed method, a model of "5" was constructed to represent various kinds of images of "5," which can also represent these minorities. Hence, some images of "6" were represented by the model of "5" and these were recognized as "5." To diminish this kind of misclassification, we should consider the relationships between the categories.

4. Conclusions

In this paper, we have proposed a novel statistical model based on the Gaussian mixture model. In the proposed method, first the elements of the feature vector are rearranged based on the covariance of the elements. A graph cut approach is used for optimization and the normalized cut criterion is adopted. Then the elements of the feature vector are classified into subsets and new random vectors are generated. Finally, a Gaussian mixture model is constructed for each subset of elements.

To confirm the effectiveness of the proposed method, experiments were carried out with the MNIST handwritten digit database. Experimental results demonstrated that the proposed method is better than the traditional method, especially when the number of sample patterns is small. The t-test was carried out and it was shown that the difference in the average accuracies of these methods is statistically significant.

In this study, we have used only the MNIST database. Applying the proposed method to various kinds of pattern recognition problems is an important future work. In addition, as we mentioned above, the Gaussian mixture model is a general statistical model and can be used in many applications. Confirming the effectiveness of the proposed method in various applications is also a future work.

Acknowledgments

This research is partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Exploratory Research, 20650024, 2008, and Grant-in-Aid for Scientific Research, 20500150, 2008.

References

[1] S. Omachi, M. Omachi, and H. Aso, "An approximation method of the quadratic discriminant function and its application to estimation of high-dimensional distribution," IEICE Trans. Inf. & Syst., vol.E90-D, no.8, pp.1160–1167, Aug. 2007.

[2] R. Gross, J. Yang, and A. Waibel, "Growing Gaussian mixture models for pose invariant face recognition," Proc. 15th International Conference on Pattern Recognition, vol.1, pp.1088–1091, 2000.

[3] G. Li, T.-Y. Leong, and L. Zhang, "Translation initiation sites prediction with mixture Gaussian models in human cDNA sequences," IEEE Trans, Knowl. Data Eng., vol.17, no.8, pp.1152–1160, 2005.

[4] C.P. Lim, S.S. Quek, and K.K. Peh, "Application of the Gaussian mixture model to drug dissolution profiles prediction," Neural Computing & Applications, vol.14, no.4, pp.345–352, 2005.

[5] S.W. Choi, J.H. Park, and I.-B. Lee, "Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis," Computers and Chemical Engineering, vol.28, no.8, pp.1377–1387, 2004.

[6] T. Chen, J. Morris, and E. Martin, "Probability density estimation via an infinite Gaussian mixture model: Application to statistical process monitoring," J. Royal Statistical Society: Series C (Applied Statistics), vol.55, no.5, pp.699–715, 2006.

[7] L. Gupta and T. Sortrakul, "A Gaussian-mixture-based image segmentation algorithm," Pattern Recognit., vol.31, no.3, pp.315–325, 1998.

[8] H. Greenspan, A. Ruf, and J. Goldberger, "Constrained Gaussian mixture model framework for automatic segmentation of MR brain images," IEEE Trans. Med. Imaging, vol.25, no.9, pp.1233–1245, 2006.

[9] P. Paalanen, J.-K. Kamarainen, J. Ilonen, and H. Kälviäinen, "Feature representation and discrimination based on Gaussian mixture model probability densities — Practices and algorithms," Pattern Recognit., vol.39, no.7, pp.1346–1358, 2006.

[10] M. Ouyang, W.J. Welsh, and P. Georgopoulos, "Gaussian mixture clustering and imputation of microarray data," Bioinformatics, vol.20, no.6, pp.917–923, 2004.

[11] H. Sahbi, "A particular Gaussian mixture model for clustering and its application to image retrieval," Soft Computing – A Fusion of Foundations, Methodologies and Applications, vol.12, no.7, pp.667–676, 2008.

[12] R. Koshiba, M. Tachimori, and H. Kanazawa, "A flexible method of creating HMM using block-diagonalization of covariance matrices," Proc. 5th International Conference on Spoken Language Processing, vol.7, pp.2947–2950, 1998.

[13] F. Sun, S. Omachi, N. Kato, and H. Aso, "Fast and precise discriminant function considering correlations of elements of feature vectors and its application to character recognition," Systems and Computers in Japan, vol.30, no.14, pp.33–42, Dec. 1999.

[14] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.8, pp.888–905, Aug. 2000.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol.86, no.11, pp.2278–2324, 1998.

[16] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.

[17] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, 2nd ed., John Wiley & Sons, 1971.

[18] J.H. Friedman, "Regularized discriminant analysis," J. American Statistical Association, vol.84, pp.165–175, 1989.

[19] F. Kimura, K. Takeshita, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol.9, no.1, pp.149–153, Jan. 1987.

[20] N. Kato, M. Suzuki, S. Omachi, H. Aso, and Y. Nemoto, "A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance," IEEE Trans. Pattern Anal. Mach. Intell., vol.21, no.3, pp.258–262, March 1999.

Masako Omachi received her B.E., Master of Information Sciences, and Doctor of Engineering degrees from Tohoku University, Japan, in 1994, 1996, and 1999, respectively. From 1999 to 2010, she was with the Faculty of Science and Technology, Tohoku Bunka Gakuen University. Since 2010, she has been with the Advanced Course of Production System and Design Engineering, Sendai National College of Technology, where she is currently an associate professor. Her research interests include pattern recognition, character recognition, and image processing. She received the MIRU Nagao Award in 2007.

Shinichiro Omachi received his B.E., M.E., and Doctor of Engineering degrees in Information Engineering from Tohoku University, Japan, in 1988, 1990 and 1993, respectively. He worked as a research associate at the Education Center for Information Processing at Tohoku University from 1993 to 1996. Since 1996, he has been with the Graduate School of Engineering at Tohoku University, where he is currently a professor. From 2000 to 2001, he was a visiting associate professor at Brown University. His research interests include pattern recognition, computer vision, image processing and parallel processing. He received the MIRU Nagao Award and IAPR/ICDAR Best Paper Award in 2007. Dr. Omachi is a member of the IEEE, the Information Processing Society of Japan, and the Japanese Society of Artificial Intelligence, among others.