Building Compact MQDF Classifier for Off-line Handwritten Chinese Characters by Subspace Distribution Sharing[1]

Teng Long and Lian-Wen Jin
School of Electronics and Information
South China University of Technology, Guangzhou 510641, P.R. China
{tenglong,eelwjin}@scut.edu.cn

Abstract

Quadratic classifier with modified quadratic discriminant function (MQDF) has been successfully applied to recognition of handwritten characters to achieve very good performance. However, for large category classification problem such as Chinese character recognition, the storage of the parameters for the MQDF classifier is huge which makes it not practical to be used in hand-held devices.

In this paper, we propose a method to build a compact MQDF classifier for handwritten Chinese characters. We use subspace distribution sharing to form a codebook for the subspaces of the MQDF classifier's parameters. Further more, we employ a two-level Euclidean distance classifier to accelerate the recognition process. Fast recognition speed and compact dictionary size make the high accuracy quadratic classifier be practical for hand-held devices.

1. Introduction

The emergence of personal hand-held devices, such as PDA's (personal digital assistants) and mobile phones, with considerable computing units and touch panels, made the handwriting recognition widely used for their limited size of keyboards. These applications gained many interests of researchers to embed the existing popular recognition approaches for handwritten characters into these hand-held devices. During the last decade, the statistical approaches have been widely developed and used for handwritten character recognition. They became the most popular methods in the literature for their simplicity and robustness [1]. Among them, the modified quadratic discriminant function (MQDF) [2] based on Bayesian decision rule has been extremely successful and makes the very important part of the reported high accuracy classifiers [3] [4]. With the power of quadratic classifier, the recognition accuracy reaches above 99% for handwritten digits and Kanji characters and above 98% for handwritten Chinese characters [3]. Although the accuracy is very high, the parameters of such kind of classifiers are usually very huge for large character set recognition such as Chinese character recognition. As the storage of the hand-held devices is usually limited at present, it is not practical to embed the MQDF classifier directly into these devices.

In this paper, we explore the method for building a compact MQDF classifier for embedded systems. In order to compress the parameters, we map the original parameter space into multiple subspaces. Then we try to find a uniform statistical model to fit the distributions of the subspaces. By sharing the distribution of multiple subspaces of the parameters, the dictionary of the MQDF classifier can be compressed greatly with a slight recognition accuracy loss. Similar technique called split vector quantization is originally developed in the automatic speech recognition (ASR) [5] [6], and also has been successfully used in compressing parameters of the continuous-density hidden Markov model (CDHMM) for handwritten Chinese character recognition [7]. From the experimental results in this paper, it is shown that even by using a shared distribution model for all subspaces of the parameters, the recognition accuracy only decreased by less than 0.2% for handwritten Chinese characters. A two-level Euclidean distance classifier is employed for coarse recognition to accelerate the recognition process. The fast recognition speed and compact dictionary size make the high accuracy MQDF classifier be practical for hand-held devices such as PDA's and Pocket PC's.

The rest of this paper is organized as follows. Section 2 reviews the MQDF classifier. In Section 3, we describe how to build a compact MQDF classifier

for handwritten Chinese characters. The experimental results on handwritten Chinese character recognition are shown in Section 4. Finally, we draw our conclusions in Section 5.

2. MQDF classifier

Based on Bayesian decision rule, which classifies the input pattern to the class of maximum a posteriori (MAP) probability out of classes, the quadratic discriminant function (QDF) is obtained under the assumption of multivariate Gaussian density for each class. The MQDF proposed by Kimura et al. [2] makes a modification to the QDF by smoothing the minor eigenvalues to improve the computation efficiency and classification performance. We review it here.

According to the Bayes rule, the a posteriori probability is computed by:

$$P(\omega_i \mid x) = \frac{P(\omega_i)p(x \mid \omega_i)}{p(x)}, \quad i = 1, \ldots, M \qquad (1)$$

where $P(\omega_i)$ is the a priori probability of class ω_i, $p(x \mid \omega_i)$ is the class probability density function and $p(x)$ is the mixture density function. As $p(x)$ is independent of class, the nominator of (1) can be used as the discriminant function for classification:

$$g(x, \omega_i) = P(\omega_i)p(x \mid \omega_i). \qquad (2)$$

Assume the probability density function of each class is multivariate Gaussian:

$$p(x \mid \omega_i) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp[-\frac{(x-\mu_i)^T \sum_i^{-1}(x-\mu_i)}{2}] \quad (3)$$

where μ_i and Σ_i denote the mean vector and the covariance matrix of class ω_i respectively. Inserting (3) into (2), taking the negative logarithm and omitting the common terms under equal a priori probabilities, the QDF is obtained:

$$g_0(x, \omega_i) = (x-\mu_i)^T \sum_i^{-1}(x-\mu_i) + \log|\Sigma_i|. \quad (4)$$

The QDF can be used as a distance metric in the sense that the class of minimum distance is assigned to the input pattern.

By K-L transform, the covariance matrix can be diagonalized as

$$\Sigma_i = \Phi_i \Lambda_i \Phi_i^T \qquad (5)$$

where $\Lambda = diag[\lambda_{i1}, \ldots, \lambda_{iD}]$ with λ_{ij}, $j = 1, \ldots, D$, being the eigenvalues (ordered in decreasing order) of Σ_i, and $\Phi_i = [\phi_{i1}, \ldots, \phi_{iD}]$ with ϕ_{ij}, $j = 1, \ldots, D$,

being the ordered eigenvectors. Φ_i is ortho-normal (unitary) such that $\Phi_i^T \Phi_i = I$.

According to (5), the QDF can be rewritten in the form of eigenvectors and eigenvalues:

$$g_0(x, \omega_i) = [\Phi_i^T(x-\mu_i)]^T \Lambda_i^{-1} \Phi_i^T(x-\mu_i) + \log|\Lambda_i|$$
$$= \sum_{j=1}^{D} \frac{1}{\lambda_{ij}}[\phi_{ij}^T(x-\mu_i)]^2 + \sum_{j=1}^{D} \log \lambda_{ij}. \qquad (6)$$

By replacing the minor eigenvalues with the mean of them

$$\delta_i = \frac{1}{D-K} \sum_{j=K+1}^{D} \lambda_{ij}, \qquad (7)$$

the MQDF is obtained as

$$g_1(x, \omega_i) = \sum_{j=1}^{K} \frac{1}{\lambda_{ij}}[\phi_{ij}^T(x-\mu_i)]^2$$
$$+ \sum_{j=K+1}^{D} \frac{1}{\delta_i}[\phi_{ij}^T(x-\mu_i)]^2 + \sum_{j=1}^{K} \log \lambda_{ij}$$
$$+ (D-K) \log \delta_i$$
$$= \sum_{j=1}^{K} \frac{1}{\lambda_{ij}}[\phi_{ij}^T(x-\mu_i)]^2 + \frac{1}{\delta_i} r_i(x)$$
$$+ \sum_{j=1}^{K} \log \lambda_{ij} + (D-K) \log \delta_i$$

$$(8)$$

where K denotes the number of dominant axes and $r_i(x)$ is the residual of subspace projection:

$$r_i(x) = \|x-\mu_i\|^2 - \sum_{j=1}^{K}[(x-\mu_i)^T \phi_{ij}]^2. \qquad (9)$$

The above utilizes the invariance of Euclidean distance:

$$d_E(x, \omega_i) = \|x-\mu_i\|^2 = \sum_{j=1}^{D}[\phi_{ij}^T(x-\mu_i)]^2. \quad (10)$$

Since the training of the QDF classifier always underestimate the patterns' eigenvalues by limited sample set, the minor eigenvalues become some kind of unstable noises and affect the classifier's robustness. By smoothing them in the MQDF classifier, not only the classification performance is improved, but also the computation time and storage for the parameters are saved.

3. Building compact MQDF classifier

For each class, the MQDF classifier needs to train and store the dominant eigenvalues and eigenvectors of the covariance matrix Σ_i. The storage problem
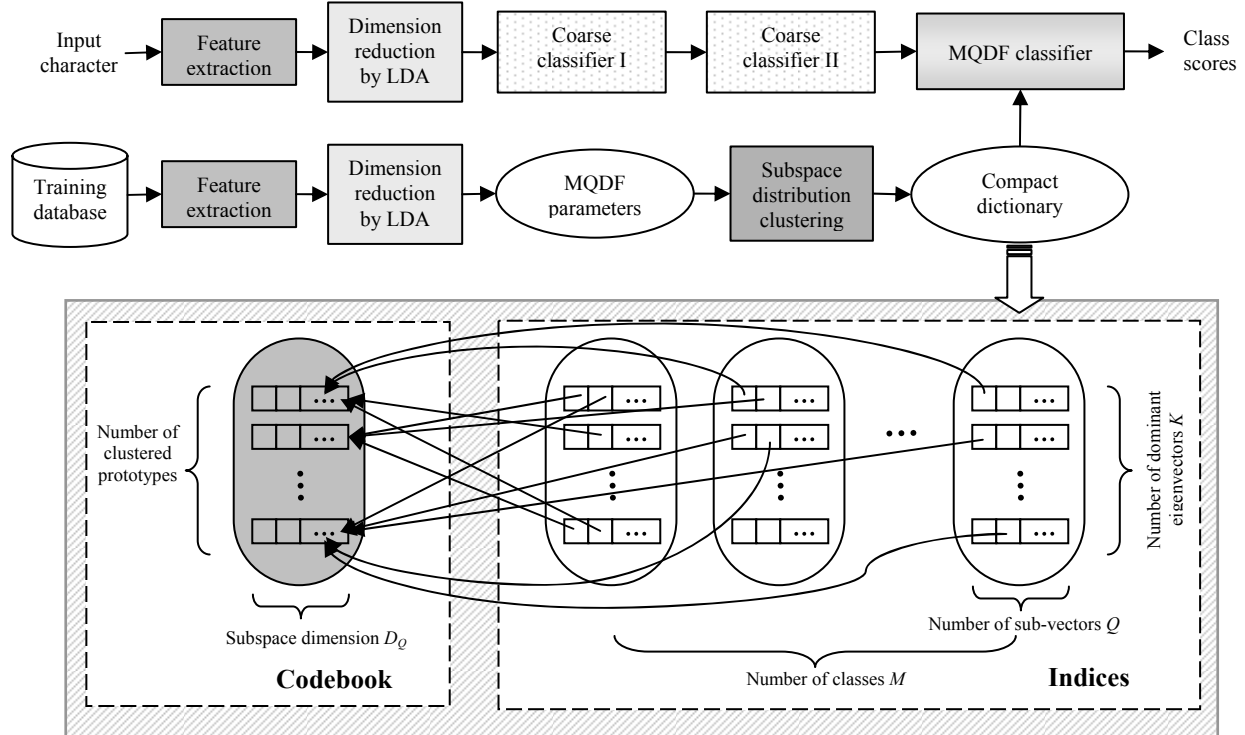
Fig. 1 Block diagram of the recognition system using the compact MQDF classifier

mainly comes from the eigenvectors of each class. From the description in Section 2, the storage of the eigenvectors Φ_i ($i = 1, \ldots, M$) is $4 \times D \times K \times M$ bytes when 4-byte floating number is used. This requires about 183MB storage space under a system setup of $D = 512$, $K = 25$, $M = 3755$. Although the dimension D of the feature vectors can be reduced by linear discriminant analysis (LDA), the eigenvectors still need about 57M bytes storage after the dimension is reduced to 160. This is very critical for the mainstream PDA and Pocket PC while their memory space is usually equal or less than 64MB.

In order to compress the eigenvectors of each class, we first map the original parameter space into multiple subspaces. This leads to a more general distribution within the subspaces. For each class ω_i, the dominant eigenvector matrix $\Phi_i^K = [\phi_{i1}, \ldots, \phi_{iK}]$ with ϕ_{ij}, $j = 1, \ldots, K$, is partitioned into subspace eigenvector matrix, i.e. each D-dimensional eigenvector ϕ_{ij} is equally partitioned into Q D_Q-dimensional sub-vectors $\phi_{ij}^1, \phi_{ij}^2, \ldots, \phi_{ij}^Q$, where $D = D_Q \times Q$.

Then we try to find a uniform statistical model to fit the distributions of all the sub-vectors. By using LBG clustering algorithm [8] in the subspaces of the parameters, the sub-vectors ϕ_{ij}^q with $i = 1, \ldots, M$, $j = 1, \ldots, K$, $q = 1, \ldots, Q$, are clustered into a small set of L prototypes. Each original subspace eigenvector is then presented by its nearest prototype. When the L is smaller than 2^r, where r is the number of bits for storing one D_Q-dimensional sub-vector, the storage of the eigenvector matrix of each class is compressed.

The building process of the prototype set is similar to the split-dimension vector quantization for speech signals [6]. Typically, the smaller quantization error reached by the clustering process, the closer approximation obtained for the original sub-vectors. This can be ensured by using more prototypes. However, the more prototypes the clustering process employs, the longer the convergence time for the clustering will be.

After the clustering process is completed, the prototypes are clustered to approximate the distribution of the subspace eigenvectors. The dictionary for the eigenvector matrices of all classes consists of two parts, the indices of the prototypes and the codebook. Fig. 1 demonstrates both the block diagram of our recognition system and the mapping of the compact dictionary.

When using two bytes for storing the indices of the prototypes, the number of prototypes can be up to

65536. However, this makes the LBG clustering algorithm consume too much time to be practical for $D \times K \times M$ samples, i.e. 15,020,000 samples when D=160, K=25, M=3755. We use the following simple scheme for clustering when using two bytes for storing the indices of the prototypes:

Step1: Select the first sub-vector as one prototype and add it to the prototype set Ω;

Step2: For each sub-vector, find the nearest prototype in the prototype set Ω, if the Euclidean distance between them is larger than a threshold δ, the sub-vector is selected as a prototype and added to Ω;

Step3: Terminate if all sub-vectors are visited, or the number of prototypes reaches 65536; otherwise, back to step 2.

By using the above simple scheme, only one run of the sub-vectors is needed for the clustering process, which saves much time for clustering large number of prototypes. The threshold δ in this scheme is chosen by experiments. Obviously, by this kind of clustering method, the minimum squared error is not ensured. Nevertheless, the experimental results have shown that such kind of approximation is effective and accurate enough for prototype building.

4. Experiments

We evaluate the performance of our compact MQDF classifier on recognition of handwritten Chinese characters. The HCL2000 database is used for both training and testing for our classifier. HCL2000 is collected by Beijing University of Posts and Telecommunications for China 863 project. It contains 3,755 frequently used Chinese characters in GB2312-80 written by 1,000 different people. We use 700 sets (labeled as xx001-xx700) for training and the rest 300 sets (labeled as hh001-hh300) for testing. The testing platform is on a PC with Pentium4 3.2G CPU and 1G memory.

The gradient feature and the decomposition method proposed by Liu et al. [10] for feature extraction are used in the experiments. For each input character image of size 64x64, we first calculate the 8 directional gradient features on each pixel by 3x3 sobel operators. After passing a Gaussian filter, the 8 directional gradient feature images are down-sampled to 8x8 by elastic meshing [9] and 512 dimensional feature vectors are extracted. Each element of the feature vector is finally transformed by $y=x^{0.5}$ to make its distribution more Gaussian-like.

Before classification, the dimension of each feature vector is reduced to 160 by LDA, which not only improves the classification performance but also arranges the dominant features for coarse classifier. Considering the MQDF is time consuming, a two-level Euclidean distance coarse classifier is employed in the recognition system. The first level coarse classifier uses first 16 dominant LDA features for classification and generates about 300 candidates for the second level coarse classifier, which uses 160 dominant LDA features and generates 10 candidates for the MQDF classifier. In order to avoid the sorting process for the candidates in the first level coarse classifier, the distance between the input character and each prototype is recorded and a histogram of the distance distribution is calculated. By accumulating the number of candidates from the lower distance in the histogram, a distance threshold is determined to give about 300 candidates by the recorded distance. The hit rate of the 10 candidates given by the two-level coarse classifier is 99.5% and the classification time is 1.3ms/char.

The MQDF classifier calculates the MQDF of the 10 candidates given by the coarse classifiers with K=25 and gives the recognition result by choosing the one which has the minimum MQDF score. The

Table 1. A summary of the open-test recognition accuracies and the memory requirements of the eigenvectors in the compact dictionary.

Sub-Vector Dimension D_Q	Clustering Method	Number of Prototypes	Mean Squared Error	Recognition Accuracy	Indices	Codebook	Total
1	LBG	256	1.20×10^{-6}	97.52%	14.3MB	1KB	14.3MB
2	LBG	256	1.05×10^{-4}	97.22%	7.16MB	2KB	7.16MB
4	LBG	256	7.60×10^{-4}	96.09%	3.58MB	4KB	3.58MB
2	$\delta = 1 \times 10^{-5}$	39201	1.33×10^{-6}	**97.53%**	14.3MB	0.30MB	14.6MB
4	$\delta = 1.2 \times 10^{-3}$	64792	1.17×10^{-4}	**97.34%**	7.16MB	0.99MB	8.15MB
8	$\delta = 2.05 \times 10^{-2}$	63154	1.36×10^{-3}	95.06%	3.58MB	1.93MB	5.51MB
Original	-			**97.53%**	-		57.3MB

performance and the memory requirements of the eigenvectors are summarized in Table 1. It is shown that only 8.15MB storage of the eigenvectors is required to reach a recognition accuracy of 97.34% for the handwritten Chinese characters, while the original uncompressed dictionary requires 57.3MB for the eigenvectors. When compressing the 4-dimensional sub-vector into one byte, the storage of the eigenvectors is greatly reduced to 3.58MB while the recognition rate only decreases by 1.4%, which shows the power of the subspace distribution sharing.

It is also noticed from the experimental results that the smaller mean squared error does not always promise the higher recognition rate. By using the threshold clustering scheme to cluster 64792 prototypes for the 4 dimensional subspaces, the mean squared error is 1.17×10^{-4}, while the recognition rate is higher than the LBG clustering for 2 dimensional subspaces with 1.05×10^{-4} mean squared error.

The recognition speed is about 4.5ms/char which promised the performance of the compact MQDF classifier on the mainstream PDA's and Pocket PC's.

5. Conclusion

In this paper, we described a method for building compact MQDF classifier. We first map the original parameter space into multiple subspaces. A uniform distribution for the subspaces is then found by the clustering process. By using a small set of prototypes clustered from the original subspaces to represent the uncompressed sub-vectors, the storage of the MQDF parameters is greatly compressed. With the reduced memory requirement of the compact dictionary and the accelerated coarse classification by a two-level coarse classifier, the MQDF classifier becomes practical to be used for the existing hand-held devices such as PDA's and Pocket PC's.

References

[1]. A. K. Jain, R. P.W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, Jan. 2000.

[2]. F.Kimura, K.Takashina, S.Tsuruoka, and Y.Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, No.1, pp. 149-153, 1987.

[3]. H. Liu and X. Ding, "Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes," *Proc. 8th ICDAR*, pp.19-23, Seoul, Korea, 2005.

[4]. C.-L. Liu, "High Accuracy Handwritten Chinese Character Recognition Using Quadratic Classifiers with Discriminative Feature Extraction," *18th Int. Conference on Pattern Recognition*, pp. 942-945, 2006.

[5]. E. Bocchieri and B. Mak, "Subspace distribution clustering hidden Markov model," *IEEE Trans. on Speech and Audio Processing*, Vol. 9, No. 3, pp.264-275, 2001.

[6]. K. W. Law and C. F. Chan, "Split-dimension vector quantization of Parcor coefficients for low bit rate speech coding," *IEEE Trans. Speech Audio Processing*, Vol. 2, pp. 443–446, July 1994.

[7]. Y. Ge and Q. Huo, "A study on the use of CDHMM for large vocabulary off-line recognition of handwritten Chinese characters," *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pp. 334-338, 2002.

[8]. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, Vol. COM-28, No. 1, pp. 84–95, 1980.

[9]. L.-W. Jin, G. Wei, "Handwritten Chinese Character Recognition with Directional Decomposition Cellular Features," *Journal of Circuit, System and Computer*, Vol.8, No.4, pp.517-524, 1999.

[10]. C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques", *Pattern Recognition*, Vol. 37, No. 2, pp. 265-279, 2004.