A New AdaBoost Algorithm for Large Scale Classification And Its Application to Chinese Handwritten Character Recognition

Qiang Fu Xiaoqing Ding Changsong Liu
Dept. of Electronic Engineering, Tsinghua University, P.R.China
State key Laboratory of Intelligent Technology and Systems
{fuq,dxq,lcs}@ocrserv.ee.tsinghua.edu.cn

Abstract

The present multiclass boosting algorithms are hard to deal with Chinese handwritten character recognition for the large amount of classes. Most of them are based on schemes of converting multiclass classification to multiple binary classifications and have high training complexity. The proposed multiclass boosting algorithm adopts the descriptive model based multiclass classifiers (Modified Quadratic Discriminant Function, MQDF) as the element classifiers, which perform multiclass classifications directly. The proposed boosting algorithm does not need to convert multiclass classifications to multiple binary classifications, and has lower training complexity than most of present multiclass boosting algorithms. So it is more suitable for dealing with large scale classification problems. The algorithm updates samples' weights according to the generalized confidence which is simple and effective. Further, in order to reduce the recognition complexity, the pruning method was performed to pick out only one best element classifier from all boosted classifiers to do the classification. Applying the proposed algorithm to Chinese handwritten character recognition on the different datasets, the recognition rate is significantly improved; meanwhile the recognition complexity is the same as the traditional MQDF classifier.

Keywords: multiclass boosting algorithm, handwritten Chinese character recognition, generalized confidence, modified quadratic discriminant function

1. Introduction

Boosting is a general framework for improving classifier's performance. It constructs multiple element classifiers according to different sample distributions, and uses the additive model to combine those element classifiers to obtain a strong classifier. In each round of iteration, it increases misclassified samples' weights and reduces right classified samples' weights so that the subsequent element classifier could give more emphasis on those misclassified samples. Adaboost is the typical boosting algorithm which has been successfully used in face detection and many other fields. It is originally designed for binary classification problems. Afterwards, it has many extensions for multiclass classification problems, such as Adaboost.M1, Adaboost.M2, Adaboost.MH, Adaboost.OC and Adaboost.ECC [1-5]. There are two key points in boosting algorithms: (1) sample weights updating algorithms; (2) element classifier algorithms. We give the brief summary on the two key points as follows.

Sample weights updating algorithms could be mainly divided into two categories: one is updating sample weights according to element classifiers' recognition rates (Discrete Adaboost); the other is updating sample weights according to samples' recognition confidences (Real Adaboost). Generally speaking, the second modus is more elaborate than the first one, so it could achieve higher recognition rate. But, the second modus need to estimate samples' recognition confidences which may be complex and difficult in some practices.

Element classifier algorithms used in the multiclass boosting algorithms could adopt two kinds of schemes: (1) using multiclass classifiers as element classifiers to perform multiclass classification directly, such as Adaboost.M1; (2) converting multiclass classifications to multiple binary classifications, and then applying multiple binary classifiers as element classifiers. Most of present multiclass boosting algorithms are based on the second scheme. There are mainly three kinds of strategies of converting multiclass classifications to multiple binary classifications: 1v1, 1vL (L means left) and output code. Adaboost.M2 and Adaboost.MH use the 1vL strategy, while Adaboost.OC and Adaboost.ECC use the output code strategy. Denote the number of training samples as N, the number of classes as C. Assuming that the training complexity is proportional to the sample times used in the training computation, the training complexity of 1v1 and 1vL strategies is approximately proportional to $N \times C$, and the training complexity of output code strategy is at least proportional to $N \times \log(C)$. For Chinese character recognition, C=3755, so 1v1 and 1vL strategies are hard to deal with it for their high training complexity. Although the training complexity of output code strategy is much

lower than 1v1 and 1vL strategies, how to construct a proper output code in the case of large scale classification has not been solved well. Above all, the schemes of converting multiclass classifications to multiple binary classifications are hard to be applied in Chinese handwritten character recognition because of the large amount of classes. So, in the proposed algorithm, we do not use the schemes of converting multiclass classifications to multiple binary classifications, while adopt multiclass classifiers as element classifiers to perform multiclass classifications directly.

The descriptive model based classifiers, such as Gaussian model based classifier, are common multiclass classifiers. The other kinds of classifiers are discriminative model based classifiers, such as decision tree, SVM and neural networks. Both discriminative model and descriptive model have the merit and shortcoming respectively [6]: Discriminative models are usually superior to descriptive models on the recognition rate, while descriptive models have stable performance than discriminative models. Furthermore, in general speaking, discriminative models have higher training complexity than descriptive models. In Chinese character recognition fields, descriptive model based classification algorithm, modified quadratic discriminant function (MQDF), have been widely used due to its promising performance, simple training scheme and relative low complexity.

The paper proposes a modified boosting algorithm which adopts the descriptive model based multiclass classifier, MQDF, as the element classier and updates samples' weights according to the generalized confidences. The algorithm boosts MQDF's performance so that the recognition rate was much higher than the traditional MQDF classifier; meanwhile the recognition complexity is same as the traditional MQDF classifier. The rest of this paper is organized as follows. In section 2, we give the modified boosting algorithm's outline which will give a summary on difference between the traditional multiclass boosting algorithms and the proposed boosting algorithm. In section 3, we give the flow and the details of the proposed modified boosting algorithm. Section 4 illustrates the pruning method used in the paper for reducing recognition complexity. Section 5 and 6 are experiments and conclusion respectively.

2. The algorithm Outline

Comparing with the traditional multiclass boosting algorithms, the proposed modified boosting algorithm has the following characteristic.

(I) The adopted element classifier is different

- Most of the traditional boosting algorithms adopt discriminative model based classifiers as element classifiers and convert multiclass classification to multiple binary classifications. So they usually have higher training complexity.
- The proposed algorithm adopts the descriptive model based multiclass classifier, MQDF, as the element classifier. It performs multiclass classification directly, and does not need to convert multiclass classification to multiple binary classifications. So, it has lower training complexity and is applicable for large scale classifications such as Chinese handwritten character recognition.

(II)The element classifiers' performance is different

- The traditional boosting algorithm's element classifiers are usually "weak learner" whose recognition rates are merely better than fifty percent.
- The element classifier used in the proposed algorithm, MQDF, is not the traditional sense of the "weak learner". It has much higher recognition rate than fifty percent. Its recognition rate has achieved about 90% on unconstrained Chinese handwritten character recognition, meanwhile above 95% on neat writing Chinese characters. So it is in fact a "strong learner". The paper shows that boosting methods are also effective for improving performance of "strong learner".

(III) Function of updating samples' weight is different

- One typical multiclass boosting algorithm which performs multiclass classifications directly is Adaboost.M1. It updates samples' weights according to element classifiers' recognition rates while ignores samples' recognition confidences, just like Discrete Adaboost. Through experiments, we find that updating samples' weights according to element classifiers recognition rates could not get good results using such strong learner as MQDF.
- We learn the idea of Real Adaboost which updates samples' weights according to samples' recognition confidences. However, it is not easy to calculate samples' recognition confidence accurately. Furthermore, the method of calculating recognition confidences in Real Adaboost is for binary classification case, it may cause problems in multiclass classification case. In the paper, we update samples' weights according to the generalized confidences which could be easily calculated. The experiments show that the proposed weight updating method is simple and effective.

After using the proposed boosting algorithm, we obtain a group of element classifiers (MQDF). Next, we select only one best MQDF classifier as the final classifier in order to reduce the recognition complexity. Finally, we get the MQDF classifier which has the same recognition complexity as the traditional MQDF classifier, meanwhile has the significant improvement on recognition correct rate.

3. The algorithm flow

We introduce Real Adaboost first. Then, we generalize it to multiclass cases and do the modification on the

confidence calculation method to obtain the proposed boosting algorithm. Denote training sample set as $\{(x_i, y_i) \mid 1 \le i \le N\}$, x_i is feature vector, y_i is the label of x_i. T is the total round number, and t is the round index.

3.1. Real Adaboost introduction

In Real Adaboost, $y_i \in \{-1, +1\}$.

(I) Initialize: t=1, $w_t^i = \frac{1}{N}$.

(II) For t=1,2,…,T

- Under the distribution w_t, estimate the following posterior probability.

$$p_t(y = 1 \mid x) \in [0,1] \tag{1}$$

- According to the formula (1), get the element classifier $h_t(x)$.

$$h_t(x) = \frac{1}{2} \log \frac{p_t(y = 1 \mid x)}{1 - p_t(y = 1 \mid x)} \tag{2}$$

- Update the samples' weight as follows.

$$w_{t+1}^i = \frac{w_t^i \exp(-y_i h_t(x_i))}{Z_t} \tag{3}$$

Z_t is the normalized factor which makes w_{t+1} as a distribution.

(III) The final decision function is $H(x)$.

$$H(x) = sign[\sum_{t=1}^{T} h_t(x)] \tag{4}$$

In Real Adaboost, $h_t(x)$ contains two meanings: its sign indicates the recognition result; its absolute value expresses the recognition result's confidence.

3.2. The modified boosting algorithm flow

In the multiclass cases, $y_i \in \{1, 2, ..., C\}$. Different from Real Adaboost, we could not use a sign to indicate a recognition result in multiclass cases. Here, we denote the t^{th} classifier as (h_t, s_t). $h_t(x)$ indicates recognition result of x, $h_t(x) \in \{1, 2, ..., C\}$; $s_t(x)$ is its generalized confidence.

(I) Initialize: t=1, $w_t^i = \frac{1}{N}$.

(II) For t=1,2,…,T

- Under the distribution w_t, train the multiclass element classifier, and get (h_t, s_t). The details of training classifier and calculating the generalized confidence are elaborated in the section 3.3 and 3.4.
- Update the samples' weight as follows.

$$w_{t+1}^i = \frac{w_t^i}{Z_t} \times \begin{cases} \exp[-s_t(x_i)] & if\ h_t(x_i) = y_i \\ \exp[s_t(x_i)] & else \end{cases} \tag{5}$$

Z_t is normalization factor which makes w_{t+1} as a distribution.

(III) The final decision function is $H(x)$. The formula (6) means that the final recognition result is the one which has the largest accumulated generalized confidence.

$$H(x) = \arg\max_r \sum_{t=1}^{T} s_t(x) \delta(h_t(x) = r) \tag{6}$$

Among it, $\delta(h_t(x) = r) = \begin{cases} 1 & if\ h_t(x) = r \\ 0 & else \end{cases}$ (7)

3.3. The training of element classifier

We use the gradient feature as the original feature whose dimension is 392 [7]. The high dimensional feature contains much non-discriminative information that will cause recognition interference. In order to eliminate the non-discriminative information and improve the recognition rate, the dimension reduction method is performed. After dimension reduction, we convert original feature vector to feature vector which are used for training MQDF classifier. So each element classifier is composed of the dimension reduction matrix and the MQDF classifier.

The most popular dimension reduction method is linear discriminant analysis (LDA). However, LDA assumes that all classes have the same covariance matrix which is not suitable for Chinese handwritten character. Further, the samples' distribution varies from round to round, so the same covariance assumption is even more unable to be satisfied. The modified heteroscedastic linear discriminant analysis (M-HLDA) does not need the assumption of different classes having the same covariance, and could extract discriminant information better. The details of M-HLDA used in the paper could be found in the paper [8]. We do the experiments using both LDA and M-HLDA for dimension reduction.

Through the dimension reduction procedures, we could get the feature dimension reduction matrix. To multiply original feature vector by dimension reduction matrix, we could get feature vector used for classification. We train the MQDF classifier [9] on those feature set. The MQDF distance can be represented as formula (8). $g_i(x)$ is the distance between feature vector x and class ω_i; m_i is the mean vector of class ω_i's samples; λ_{ij} and φ_{ij} are the j^{th} eigenvalue (in descending order) and its corresponding eigenvector of the class ω_i's covariance matrix respectively; q is the number of dominant principal axes; σ is a constant. The parameters of formula (8) are estimated using maximum likelihood (ML) framework.

During classification, MQDF calculates distances between the test sample and each class according to formula (8), and selects N classes with minimum distances as candidate results. The N candidate results are arranged in increasing order according to their distances. The candidate results' distances would be used to calculate the generalized confidence, which would be detailed in the following section.

$$g_i(x)=\frac{1}{\sigma^2}\{\|x-m_i\|^2-\sum_{j=1}^{k}(1-\frac{\sigma^2}{\lambda_{ij}})[\varphi_{ij}^T(x-m_i)^2]\}$$
$$+\sum_{j=1}^{k}\log\lambda_{ij}+(d-k)\log\sigma^2 \qquad i=1,2,\ldots C \tag{8}$$

3.4. The generalized confidence

According to Real Adaboost, the recognition confidences could be calculated as follows in multiclass cases.

$$s_t(x)=\frac{1}{2}\log\frac{p_t(h_t(x)\,|\,x)}{1-p_t(h_t(x)\,|\,x)} \tag{9}$$

Among it, $h_t(x)$ is the recognition result outputted by the multiclass classifier; $p_t(h_t(x)|x)$ is the estimation of the recognition result's posterior probability under the distribution w_t. In the multiclass cases, $p_t(h_t(x)|x)$ of some samples may be smaller than 0.5 so that $s_t(x)$ would be negative. According to the weights updating procedure, those samples whose $s_t(x)$ are negative, will increase their weights if their recognition results are correct, and decrease their weights if their recognition results are incorrect. Obviously, that is unreasonable and causes contradiction with the boosting motivation. In order to avoid the contradiction, the recognition confidence $s_t(x)$ should be ensured as nonnegative, but formula (9) could not provide that guarantee. Furthermore, estimating the recognition results' posterior probability is complex and difficult in the practice so that it is not convenient to using formula (9) to calculate recognition confidences.

In the paper, the generalized confidence is introduced as the substitution of the recognition confidence calculated by formula (9). On the one hand, the generalized confidence is always a positive value. On the other hand, it could be easily calculated. The experiments show that updating sample weights according to the generalized confidences is simple and effective.

The generalized confidence's definition is as follows: The classifier outputs $h(x)$ as the recognition result of the sample x. If there exists function $e(h(x)|x)$, and a monotonically increasing function $f(\bullet)$, which satisfy formula (10), then $e(h(x)|x)$ is called a generalized confidence. The generalized confidence used in the paper is as formula (11) [10]. Among it, $g_k(x)$ is the distance corresponding to the k^{th} candidate recognition result of x.

$$e(h(x)\,|\,x)=f(p(h(x)\,|\,x)) \tag{10}$$

$$s(x)=e(h(x)\,|\,x)=1-\frac{g_1(x)}{g_2(x)} \tag{11}$$

3.5. Section summary

The training flow could be summarized as follows: For each round, we obtain dimension reduction matrix of LDA or M-HLDA and MQDF classifier under the current samples distribution. Then, update samples weights according to the generalized confidences and do the next iteration under the new distribution. For classification, each sample needs to be recognized T times. The t^{th} recognition procedure is: apply the t^{th} dimension reduction matrix on original feature vector to obtain feature vector, then use the t^{th} MQDF classifier to recognize the feature vector, get its recognition result and the corresponding generalized confidence. With the T recognition results, we select the class which has the largest accumulated generalized confidence as the final recognition result.

4. Pruning the element classifiers

Through the boosting procedure, we obtain T element classifiers. Each element classifier is composed of a LDA or M-HLDA dimension reduction matrix and a MQDF classifier. The present popular algorithm for Chinese handwritten character recognition is LDA+MQDF, which has the same recognition complexity as one element classifier in the proposed boosting algorithm. So, the proposed boosting algorithm's recognition complexity is T times of the traditional LDA+MQDF algorithm. In order to keep the same recognition complexity as the traditional algorithm, we select only one best element classifier as the final classifier to do recognition.

Denote the t^{th} element classifier's recognition rate on the training set as CR_t. The selected element classifier's index satisfies the formula (12).

$$index=\arg\max_{1\le t\le T}\{t\,|\,(CR_t-CR_{t-1}\ge Th)\,\&\,(CR_t>CR_i,\,if\ t>i)\} \tag{12}$$

5. Experiment

5.1. Chinese handwritten character database

In the paper, we use two Chinese handwritten character databases for experiments. One is HCL2000 database, the other is THOCR-HCD database. Both of them contain 3,755 Chinese character classes of GB2312-1980. The HCL2000 database contains 1,000 sets samples written by 1,000 different people, and every set contains 3,755 character samples. We use 700 sets (labeled as xx001-xx700) for training and the rest 300 sets (labeled as hh001-hh300) for testing. The THOCR-HCD database is divided into 10 subsets which are marked as good, medium or bad according to the quality of the samples. It is detailed as table 1. In our experiment, HCD4 and HCD9 are used as

testing set; others are used as training set. Some samples are shown as figure 1 and figure 2.

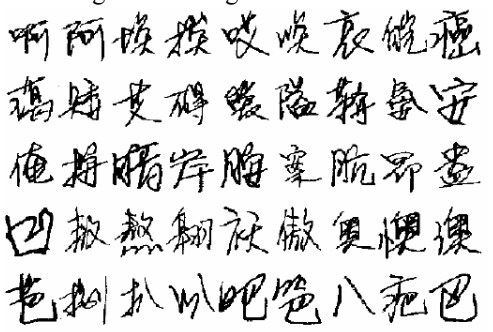

Figure 1. Samples of HCL2000 database

Table 1. THOCR-HCD database subset information

Subset	Samples	Quality
HCD-1	100×3755	good
HCD-2	500×3755	good
HCD-3	107×3755	medium
HCD-4	100×3755	medium
HCD-5	300×3755	medium
HCD-6	300×3755	medium
HCD-7	300×3755	medium
HCD-8	100×3755	bad
HCD-9	20×3755	bad
HCD-10	172×3755	bad

Figure 2. Samples of HCD database

5.2. Experiment results

Using LDA+MQDF as the element classifier, we use the proposed boosting algorithm to train 40 rounds on THOCR-HCD database, and get 40 element classifiers. The recognition rates on the training set and the two testing sets are shown in figure 3, figure 4 and figure 5. In each figure, there are two curves which express the recognition rates of the T element classifiers' ensemble and the T^{th} individual element classifier respectively.

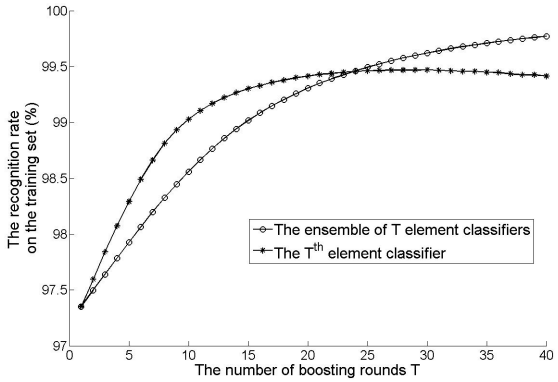

Figure 3. The recognition rate on THOCR-HCD training set

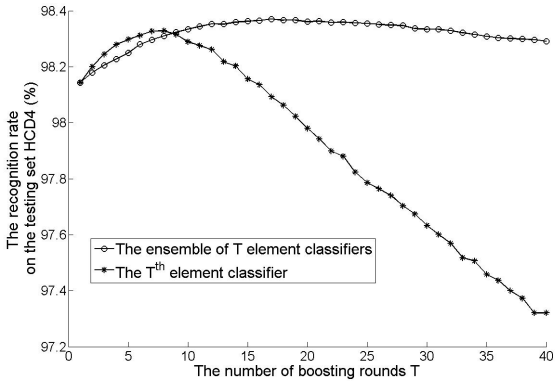

Figure 4. The recognition rate on THOCR-HCD testing set1 (HCD4)

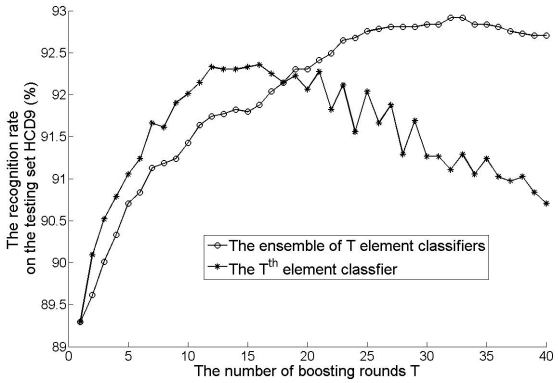

Figure 5. The recognition rate on THOCR-HCD testing set2 (HCD9)

The figures show the following points: (1) In the training set, the recognition rates of the ensemble classifiers increase persistently with the number of rounds increasing, while the individual element classifiers'

recognition rates increase at the beginning and then decrease a little at the end. (2) In the testing sets, the recognition rates of the ensemble classifiers and the individual element classifiers are all increase at the beginning and then decrease. After the peak, the recognition rates of the ensemble classifiers decrease very slowly while the individual element classifiers' very fast. (3) The peak recognition rate of the ensemble classifiers is higher than that of the individual element classifiers. Doing the experiments on HCL2000 database, the results have the similar characteristics. Although the ensemble classifier has the better performance on the recognition rate than the individual element classifier, its recognition complexity is much higher than that of the individual element classifier. So, we select the best boosted individual element classifier according to the formula (12) as the final classifier. For example, for THOCR-HCD database, we select the 9^{th} round of element classifier as the final classifier.

Besides, we use M-HLDA+MQDF as the element classifier to perform the experiments.

The recognition results of different algorithms are given in table 2. Among it, the boosted LDA+MQDF and the boosted M-HLDA+MQDF are the selected individual LDA+MQDF and M-HLDA+MQDF element classifier respectively. All of listed algorithms have the same recognition complexity. The experiments illustrate that the proposed boosting algorithm is effective no matter using LDA or M-HLDA as the dimension reduction algorithm. The boosted M-HLDA+MQDF has the highest recognition rate. Comparing with the traditional LDA+MQDF, its relative error reductions are 22.4%, 15.1% and 35.4% respectively.

Table 2. The algorithms' recognition rate comparison

Algorithm	The recognition rate (%)		
	HCL2000 Testing set	HCD Testing set1	HCD Tesing set2
LDA+MQDF	98.53	98.14	89.29
Boosted LDA+MQDF	98.75	98.29	92.01
M-HLDA+MQDF	98.67	98.28	91.40
Boosted M-HLDA+MQDF	98.86	98.42	93.08

6. Conclusion

The main characteristic of the proposed algorithm is summarized as follows:

- Adopt descriptive model based multiclass classifier as element classifier which does multiclass classification directly and does not need to convert multiclass classification to multiple binary classifications. It has much lower training complexity than most of present multiclass boosting algorithms and is more suitable for large scale classification, such as Chinese handwritten character recognition.
- Use the generalized confidence to update samples' weights. The experiments show that the method is simple and effective.
- Apply the boosting algorithm to Chinese handwritten character recognition for the first time.
- The experiments on the different Chinese handwritten character databases illustrate that the proposed algorithm achieved significant improvement than traditional algorithms, especially on those bad quality samples. Meanwhile, it keeps the same recognition complexity as traditional algorithms.

References

[1] Friedman J, Hastie T, Tibshirani R, "Additive logistic regression: a statistical view of boosting", *The Annals of Statistics,* 2000, 28(2):337-407.

[2] Schapire R E, Singer Y, "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, 1999, 37 (3):297-336.

[3] Freund Y, Schapire R E, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences*, 2006, 55(1): 119-139.

[4] Guruswami V Sahai, "A. Multiclass learning, boosting, and error-correcting codes", *Proceedings of the twelfth Annual Conference on Computational Learning Theory*, Santa Cruz, USA: ACM, 1999, 145-155.

[5] Schapire R, "Using output codes to boost multiclass learning problems", *Proceedings of the fourteenth International Conference on Machine Learning*, Nashville, USA: Morgan Kaufmann Publishers Inc, 1997, 313-321.

[6] Liu C L, Fujisawa H, "Classification and learning for character recognition: comparison of methods and remaining problems", *Proceedings of the First IAPR TC3 Workshop on Neural Networks and Learning in Document Analysis and Recognition*, Seoul, Korea: IEEE, 2005, 1-7.

[7] Liu H L, Ding X Q, "Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes", *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, Seoul, Korea: IEEE, 2005, 19-25.

[8] Liu H L, Ding X Q, "Improve handwritten character recognition performance by heteroscedastic linear discriminant analysis", *the Eighteenth International Conference on Pattern Recognition*, Hongkong, China, 2006, 880-883.

[9] Kimura F, Takashina K, Tsuruoka S, "Modified quadratic discriminant functions and its application to Chinese character recognition", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 1987, 9(1): 149-153.

[10] Lin X F, Ding X Q, Chen M, "Adaptive confidence transform based classifier combination for Chinese character recognition", *Pattern Recognition Letters*, 1998, 19 (10):975-988.