Combination of Multiple Classifiers for Handwritten Word Recognition

Wenwei Wang, Anja Brakensiek
Department of Computer Science
Gerhard-Mercator-University Duisburg
Bismarckstr. 90, 47057 Duisburg, Germany
{wwwang,anja}@fb9-ti.uni-duisburg.de

Gerhard Rigoll
Institute for Human-Machine Communication
Technical University of Munich
80290 Munich, Germany
rigoll@ei.tum.de

Abstract

Because of large shape variations in human handwriting, recognition accuracy of cursive handwritten word is hardly satisfying using a single classifier. In this paper we introduce a framework to combine results of multiple classifiers and present an intuitive run-time weighted opinion pool (RWOP) combination approach for recognizing cursive handwritten words with a large size vocabulary. The individual classifiers are evaluated run-time dynamically. The final combination is weighted according to their local performance. For an open vocabulary recognition task, we use the ROVER algorithm to combine the different strings of characters provided by each classifier. Experimental results for recognizing cursive handwritten words demonstrate that our new approach achieves better recognition performance and reduces the relative error rate significantly.

1. Introduction

For the problem of off-line handwriting recognition, feature extraction methods and classification techniques were intensively studied in the past several decades. Many recognition methods have been proposed, but none of them can reach a totally satisfactory solution of the problem.

In recent years some multiple classifier combination techniques were proposed to improve handwritten character recognition performance and have shown promising results by different researchers. Based on different results representation of the individual classifiers, diverse combination methods are available now. Most existing approaches are limited only to the case of voting differences of individual classifiers at single symbol level. Because of the complicated interaction between segmentation and recognition, optimal voting at the symbol-string level is still an open problem.

In this paper we focus on the principle of multiple-classifier combination and introduce a framework to summarize some most-used combination methods and give a hint for integration of new approaches. An intuitive run-time weighted opinion pool approach is presented which evaluates the individual classifiers run-time dynamically and performs weighted voting according to the local performance of each classifier. Another topic of this paper is the combination of multiple classifiers at the level of symbol string. We use an improved ROVER algorithm [3] to solve this problem.

In the following section we introduce a framework to express the principle of multiple classifier combination. Then we present in Section 3 the RWOP approach. In Section 4 we introduce the ROVER algorithm and some adaptations for handwritten word recognition. We provide experimental results in Section 5 and draw some conclusions in Section 6.

2. A framework for combining multiple classifiers

Multiple classifier combination is a technique of combining the decisions of different classifiers which are trained to solve the same problem, but make different errors. A proper combination of multiple classifiers should produce more reliable recognition results than any of the individual classifiers. Many combination methods have been proposed in the last decade [4, 5, 6, 12]. Most of them fall into the category of late decision-fusion, where the recognition results from different classifiers are combined without further evaluation of the original representation of the underlying objects. Thus the combination method is strongly affected by how much information the individual classifiers provide [12]. In the literature these methods are usually discussed separately. By viewing the combination problem as a special classification problem, we introduce a unified framework to encompass some of the most-used combination methods.

From: IWFHR 2002, IEEE Computer Society, New York

2.1. A general discriminant function of decision combination

The multiple classifier combination makes the decision which class from a set of candidate classes is the final result in the most reliable sense. The decision combination can be considered as a special classification problem. The combining system should rescore each different single decision. The decision with the best score is decided as the final result. Thus defining a discriminant function for each output of individual systems and making a final decision based on the discriminant functions is of much interest. Such an approach should give a better understanding about the existing combination methods and give ideas to integrate new methods. Many of existing combination methods, such as majority voting, weighted majority voting, borda count, linear combination of confidence value and combination neural network, proceed just in this manner, this means that they make the final decision based on setting up a set of discriminant functions.

Each output of all classifiers (including the case that a single classifier provides several outputs) is labeled linearly increasing from 1 to K (where K denotes the number of modules). From now on, an output is called a module. Classifier i is denoted as e_i, module j is denoted as m_j. The class of m_j is denoted as $C(m_j)$. For the module m_j, the discriminant function is defined as:

$$D_j = f(V_j, G_j) \tag{1}$$

which is a generalized function of two factors: the item V_j is a generalized vote factor, which computes the contributions of individual candidates. The item G_j stands for a general goodness fit score which characterizes the performance of each module. Other factors (for example context score) can be also absorbed into the discriminant function when these informations are available. The two factors V_j and G_j can be given a quite different form. Xu *et al.* [12] categorized the existing classifiers into three classes according to their result representation. How to define V_j and G_j depends upon what information the individual classifiers provide.

Having set up a discriminant function, the final decision rule can then be simply expressed as:

$$E(x) = C(m_j) \quad \text{if} \quad D(j) \geq D(i), \forall i \neq j \tag{2}$$

The combining system picks the class of module m_j which has the highest score as it's final decision.

The final discriminant function (Eq.(1)) can be quite generalized. For example Lam *et al.* [7] use a MLP neural network [7] for the combination task, the output O_j of the MLP can be viewed as the discriminant function D_j. In the following it will be shown that many of the existing most-used combination methods can be expressed as a weighted sum of the voting factor. In this case, the Eq.(1) can be simplified as following:

$$D_j = \sum_i G_i \delta_{ij} V_i \tag{3}$$

while a simple vote count is defined as:

$$CV_j = \sum_i \delta_{ij} V_i \tag{4}$$

δ_{ij} is an indicator contribution function which is defined as:

$$\delta_{ij} = \begin{cases} 1 & \text{if } C(m_i) = C(m_j) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

2.2. Majority Voting and Borda Count method

In this case each module m_j is associated only with a single label. The Majority Voting method select the class which gets more votes than any other class. Each candidate module has the same vote and weight which are defined as:

$$V_j = 1, \quad G_j = 1/K \quad 1 \leq j \leq K \tag{6}$$

The vote count CV_j of module m_j is accumulated across all modules.

$$CV_j = \sum_{i=1}^{K} \delta_{ij} \tag{7}$$

The discriminant function for the Majority Voting can be defined as:

$$D_j = \sum_{i=1}^{K} \delta_{ij}/K \tag{8}$$

In this case, D_j is also called occurrence frequency of class $C(m_j)$.

The Weighted Majority Voting and the Borda Count method can be seen as extension of Majority Voting method. The Weighted Majority Voting method assigns each candidate module different weight according to it's performance. Most of Weighted Majority Voting methods determine a fixed weight for each candidate with a set training data. Lam *et al.* [7] proposed a weighted majority vote method with weights derived from a genetic search algorithm and from Bayesian formulation.

The Borda Count method for ranked lists combination can be considered as a generalized form of majority voting method. Each classifier provides a list of ranked results. According to the Borda Count method [4], the different modules in the combining system should have a different vote, called Borda Count. The Borda Count B_j of modules m_j can be placed in a look-up table. Setting $V_j = B_J$ and $G_j = 1$, the discriminant function D_j is calculated according to

$$D_j = \sum_{i=1}^{K} \delta_{ij} B_i \tag{9}$$

2.3. Combination on Measurement Level

In this case each module is associated with a label and a real value indicating the recognition confidence . We assume that the different confidence measures utilized by different classifiers can be normalized to the range of [0..1] in a probability sense.

The assignment of vote should takes confidence value factor P_j into consideration. Following are some examples of how to define discriminant functions for some combination methods on Measurement Level.

2.3.1 Max rule

Setting $V_j = P_j$ and $G_j = 1$, the discriminant function is defined as: $D_j = P_j$.

2.3.2 Sum rule

Setting $V_j = P_j$ and $G_j = 1$, the discriminant function of module m_j is calculated as:

$$D_j = CV_j = \sum_{i=1}^{K} \delta_{ij} P_i \tag{10}$$

This is also called the Linear Confidence Accumulation (LCA) method [5].

2.3.3 Weighted Sum rule

Setting $V_j = P_j$ and determining the weight score G_j according to it's performance, the discriminant function can be defined as:

$$D_j = \sum_{i=1}^{K} G_i \delta_{ij} P_i \tag{11}$$

3. Run-time weighted opinion pool approach

Our final decision rule for recognition is based on the computation of the discriminant function D_j according to Eq.(11), where G_j is the goodness fit value of the jth module and P_j is the confidence score provided by the individual classifiers. Finally, the class $C(m_j)$ is selected as winning class that produces the largest value of D_j according to Eq.(2). Differing from other weighted sum methods, our combination method determines the weights G_j with an intuitive run-time approach which is thus called run-time weighted opinion pool (RWOP).

For the concrete implementation of the discriminant function, the system goodness fit score G_j can be established in different ways, while the selection of V_j is relative simple. For example Lam *et al.* [7] applied a genetic search algorithm and Bayesian formulation to get weights for individual classifiers. Some other works addressed the problem of weight selection by defining an objective function and training the weights with an optimization technique [1, 8]. But these methods need a large size of training data which should also be representative for the test data set. If the requirement is not available, the derived weights suffer from the overfitting problem and lead to decreased performance. Especially if the number of classes is high, for example in recognition tasks with large size vocabulary, such training methods are infeasible. Furthermore, different recognition engines may perform differently on different data subsets, even when we assume that they have similar performance for the total data set, because different data subsets may come from different writers, different pages or text lines. Thus we think the goodness fits of individual recognizers should be evaluated dynamically at combination run-time.

The dynamic evaluation of the goodness fits of individual recognizers can be implemented in a simple manner. From the total test data, every N units (for recognition tasks with a fixed lexicon, the unit is word with $N = 50$, for a context-free task the unit is character with $N = 200$) are grouped into a subset. The kth subset contains the [kN+1, (k+1)N] elements of the total dataset. For each recognizer module, the system records two local hits (S and T) in a subset at run-time. The count S is accumulated among the current subset, whereas T is accumulated among the current and last subset. Although the correctness of the overall decision is not known yet, the final result can be considered as an estimate of the true class index. The number of local hits of module j is incremented, if $C(m_j)$ hits the total result μ, which comes out from the combined evaluation of all modules.

$$\text{inc}(S_j, T_j) \qquad \text{if} \qquad C(m_j) = \mu \tag{12}$$

The goodness fits of the individual modules are calculated according to:

$$G_j = T_j/(N + n) \tag{13}$$

n is the unit count in the current data subset. After a run with N combination events , the S is assigned to T and a new iteration is begun. Initially, the local hits and goodness fits of the individual module are all set equal (N and 1 respectively). Figure 1 illustrates this dynamic weighted voting method.

4. Combining multiple strings with the ROVER algorithm

Most available combination methods concentrate on the decision combination from a set of single symbols (character, numeric or word as unit). If the individual classifiers

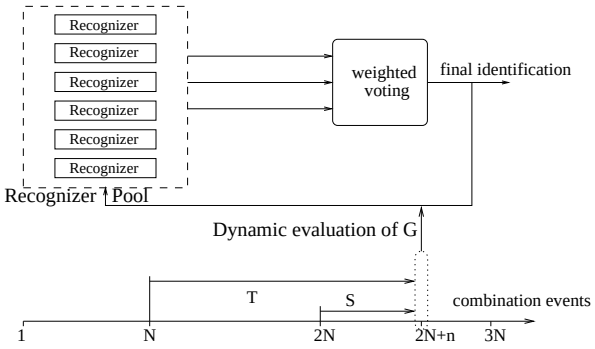

Figure 1. Dynamical evaluation of goodness fit score and weighted voting

provide a hypothesis in the form of a string of symbols, and in the hypothesis there may be not only substitution errors but also insertion and/or deletion errors, the existing methods can not be applied directly. In the literature several works have addressed this problem with preliminary result [11, 13]. We use the ROVER algorithm to deal with this problem.

4.1. Introduction to the ROVER algorithm

ROVER (Recognizer Output Voting Error Reduction) was developed by J. Fiscus of NIST [3]. Originally it aims to reduce word error rates for automatic speech recognition by combining multiple speech recognizers. The ROVER system consists of two modules. The first module is an alignment module which uses an iterative procedure to align more than two strings and build a single word transcription network (WTN). The second module is a scoring module which rescores each word at each node of the WTN and selects the word with the best score as the final decision of this node.

The alignment module treats the first WTN as the base WTN and aligns the second WTN with it using the dynamic programming alignment protocol. The base WTN is augmented with word transition arcs from the second WTN accordingly. The composite WTN is then aligned with the third WTN resulting in a new combined WTN. This process is iteratively applied for until all inputs have been combined into a single composite WTN.

Once the composite WTN has been generated from the initial system outputs, the composite WTN is searched by the scoring module to select the best scoring word sequence. Details of ROVER is described in the paper of J. Fiscus of NIST [3].

4.2. Modifications to the rover algorithm

To adopt the ROVER algorithm to the recognition of handwritten words (as strings of characters), we have taken some problem specific factors into consideration and reimplemented the ROVER algorithm with the following modifications.

4.2.1 Rescoring the combination order

It has been noticed that the composite WTN is to some extent affected by the order in which the WTNs are combined. We think that for the combination of multiple character-string recognizers, the length of the string from a recognizer plays an important role in the combination process. A correct recognition must have at first correct length. It is thus reasonable to assume that the most candidates have the same length of string, despite the possible difference in their content. Based on this consideration the candidates will be given a combination order score which consists of the length occurrence count and the goodness fit score of the recognizer.

The occurrence count L_j of the length of the string m_j is defined as $L_j = \sum_i^K \Delta_{ij}$. The indicator function Δ_{ij} is defined as:

$$\Delta_{ij} = \begin{cases} 1 & \text{if} \quad length(m_i) = length(m_j) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

The combination order score is simply the sum of L_j and G_j:

$$CS_j = L_j + G_j \tag{15}$$

The candidates will be sorted according to the combination order score. Then the assignment module proceeds in this order.

4.2.2 Dynamic voting

After the alignment process, the voting is carried out at each correspondence set with the weighted voting algorithm introduced in Section 3. The number of local hits of a module and it's goodness fit score is evaluated at character level.

5. Experiments

5.1. Building multiple classifiers based on HMM

Because of the power of Hidden Markov Model (HMM) technique, we decided on the combination of homogeneous recognizers based on HMMs. In our previous works [2, 10] some HMM-based handwriting recognition methods, including feature extraction and modeling techniques are described. In HMM-based systems many parameters have to

be selected and tuned manually through experiments. Especially the choice of features may perform differently on different data regions. Thus building multiple classifiers and combining them into an overall system is a reasonable method.

For handwritten word recognition, the word baseline is obviously an important feature, but high accurate baseline detection is very difficult. Feature extraction methods which are dependent or independent on baseline detection have thus their advantage and disadvantage. We use different sliding window zone coding methods. One method is word baseline dependent, which is described in paper [10]. The other method is word baseline independent dynamic zone coding (see paper [2] for detail, except for that no baseline dependent feature is used in the new system). The selection of the number of zones in the sliding window and whether using zone coding directly or using a transformation method, e.g. cosine transformation as feature vector leads also to different results. In our system discrete HMMs are used. For vector quantization the k-means algorithm is the most used method. With different codebook size of vector quantization, one can get different recognition results. In total, 6 classifiers (denoted as E1 , E2, ... E6) based on HMM approach are selected to be combined. Their characteristics are listed in the Table 1.

Each HMM based classifier outputs top two hypotheses (denoted as H1 and H2) and corresponding likelihood probabilities (denoted as $p1$ and $p2$). A confidence value P is transformed from the likelihoods with:

$$P = (2 * p1 - p2)/(p1 + p2) \tag{16}$$

If the classifier outputs only one hypothesis, P is set to 1. If H1 = H2, P is set to 0.75.

Table 1. Characteristics of individual classifiers

Profile	E1	E2	E3	E4	E5	E6
(word) recog. rate, 1120 words lexicon	97.04	96.59	96.86	96.95	97.00	97.13
(word) recog. rate, without lexicon	52.85	49.17	50.50	50.36	50.96	54.68
(char) recog. rate, without lexicon	86.99	83.05	83.50	83.59	84.23	90.67
word baseline	dep.	dep.	dep.	ind.	ind.	ind.
cosine trans.	yes	no	no	yes	yes	no
num. of zones	4	5	6	5	6	5
codebook size	180	200	256	220	256	256

5.2. Experimental results

To compare the proposed RWOP combination approach with other combination methods and verify application of the ROVER algorithm, experiments on the recognition of cursive handwritten words have been carried out in two different contexts: recognition using a lexicon of 1120 words and character recognition without lexicon. In the fixed lexicon mode, the RWOP combination method is used to make final decision, whereas in the lexicon-free mode the ROVER algorithm with RWOP extension is applied. The RWOP combination method are compared with some most used combination methods: Majority Voting (MV), Weighted Majority Voting (WMV) and the Linear Conference Accumulation (LCA).

The experiments have been performed on a handwritten word database which consists of about 9600 words written by six different writers. 75% of the data is taken as training set, the other 25% (2231 words, 10783 characters) as testing set. Details about the database are described in paper [10].

In the fixed lexicon mode, six classifiers are combined. The best matched word identification along with the confidence value from the individual classifiers (E1, E2,..., E6) are evaluated for combination. The recognition rates of individual classifiers are listed in the Table 1. The recognition rates achieved by different combination methods are reported in Table 2. The Run-time Weighted Opinion Pool (RWOP) method achieves a recognition rate of 98.03%, which outperforms the other combination methods and reach a 31% relative error reduction compared with the best individual recognizer.

Table 2. Word recognition rates with a lexicon of 1120 words

MV	WMV	LCA	RWOP
97.80	97.94	97.89	98.03

In the context-free mode, the same six classifiers are combined. The individual classifiers provide the best matched string of characters as the identification of the input word image. The results of individual classifiers are listed in Table 1. The results of combination systems are listed in Table 3. The ROVER algorithm achieves a significant performance gain for combination of multiple strings. The ROVER algorithm with the RWOP extension outperforms other combination methods and achieves a 13% and 35% relative error reduction at word and character level respectively against the best single classifier.

Table 3. Recognition rates without lexicon

	MV	WMV	LCA	RWOP
Word level	59.36	60.25	60.07	60.47
Character level	92.24	93.15	92.92	93.94

The NIST data set comes from the CD-ROM [9] of

"NIST Form-Based Handprint Recognition System (Release 2.0)". On the preamble fields of ten sample HSF forms (written by ten different persons), the position of each word are manually labeled. A program extracts the words using these labels and gets in total 514 words. The recognition lexicon for the task has a size of 38 words. The experimental result on the NIST data set is reported in Table 4 and Table 5. The RWOP combination method (93.97%) outperforms the best individual recognizer (91.25%) with a 31% relative error reduction.

Table 4. Word recognition rates of individual classifiers for NIST sample data

E1	E2	E3	E4	E5	E6
90.66	83.85	84.82	86.19	87.55	91.25

Table 5. Word recognition rates of combination systems for NIST sample data

MV	WMV	LCA	RWOP
91.62	93.00	93.39	93.97

6. Conclusion and outlook

A unified framework for multiple classifier combination is introduced and an intuitive run-time weighted opinion pool approach is proposed. The combining system evaluates the performance of the individual modules in the runtime and uses the goodness fit score to weight the final voting. The new method can be applied in conjunction with majority voting, Borda Count and confidence value averaging methods. For the complicated combination of multiple string recognizers, the ROVER algorithm is applied and some adaptations are proposed. Promising results have been achieved with our proposed methods.

References

[1] K. Al-Ghoneim and B. V. K. V. Kumar. Unified decision combination framework. *Pattern Recognition*, 31(12):2077–2089, 1998.

[2] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, and G. Rigoll. Performance Evaluation of a New Hybrid Modeling Technique for Handwriting Recognition Using Identical On-Line and Off-Line Data. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, pages 446–449, Bangalore, India, 1999.

[3] J. Fiscus. A Post-Processing System to Yield Reduced Error Word Rates: Recognizer Output Voting Error Reduction (ROVER). In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354, ?, ?, 1997.

[4] T. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

[5] Y. Huang and C. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(1):90–94, 1995.

[6] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(3):226–239, Mar. 1998.

[7] L. Lam, Y.-S. Huang, and C. Suen. Combination of Multiple Classifier Decisions for Optical Character Recognition. In H. Bunke and P. Wang, editors, *Handbook of Character Recognition and Document Image Analysis*, pages 79–101. World Scientific Publishing Company, 1997.

[8] X. Lin, X. Ding, M. Chen, R. Zhang, and Y. Wu. Adaptive Confidence Transform Based Classifier Combination for Chinese Character Recognition. *Pattern Recognition Letters*, 19(10):975–988, 1998.

[9] M.D.Garris, J.L.Blue, G.T.Candela, P.J.Grother, S.A.Janet, and C.L.Wilson. NIST Form-Based Handprint Recognition System (Release 2.0). Technical report, NIST Internal Report 5959 and CD-ROM, January 1997.

[10] W. Wang, A. Brakensiek, A. Kosmala, and G. Rigoll. HMM based High Accuracy Off-line Cursive Handwriting Recognition by a Baseline Detection Error Tolerant Feature Extraction Approach. In *Proceedings International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 209–218, Amsterdam, Netherlands, 2000.

[11] X. Wang, V. Govindaraju, and S. Srihari. Multi-experts for Touching Digit String Recognition. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, pages 800–803, Bangalore, India, 1999.

[12] L. Xu, A. Krzyzak, and C. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Syst. Man Cybern.*, 22(3):418–435, 1992.

[13] X. Ye, M. Cheriet, and C. Y. Suen. A Framework of Combining Numeric String Recognizers. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, pages 716–720, Seattle, USA, 2001.