Multiple Expert Decision Combination Approach for Character Recognition

Te-Wei Chiang
Computer Center
Chihlee Institute of Commerce
313, sec 1, Wen-Hua Rd, Panchiao
ctw@mail.chihlee.edu.tw

Jeng-Ping Lin
Computer Center
Chihlee Institute of Commerce
313, sec 1, Wen-Hua Rd, Panchiao
jplin@mail.chihlee.edu.tw

Abstract

In an attempt to achieve an agreeable decision, it is crucial to coordinate the opinions provided by different experts, whose expertises usually induce dissimilar conclusions. Classification problems and character recognition problems are one of the well-known instances. These problems need to apply different features to distinguish one from another. However, different features usually provide different suggestions. To avoid this kind of inconsistency, one can apply voting methods or weighting methods to generate a final decision compulsively. Nevertheless, these methods are flawed inherently. Taking voting methods for example, it is not always adequate to resort to the majority since the opinion from an authority sometimes is more credible than those from several ordinary experts. On the other hand, traditional weighting methods that assign constant weight to each expert is insufficient because experts can not only be justified according to the domains they specialize in, but also the targets they treat with. Based on the above observations, we propose a new approach, named MEDC, to support a combined decision among multiple experts. It takes advantage of the philosophy underlying neural networks and refines the traditional weighting method to achieve our goal.

Keywords: multiple experts、classification problem、character recognition、decision support system、decision combination.

1. Introduction

In the domain of classification or character recognition, no single feature can be applied individually to achieve high recognition rate. This is because every feature not only has its advantage, but also has its weakness. Therefore, varying features need to be applied simultaneously to raise the recognition rate. Without loss of generosity, we can regard each feature as an expert. Since different experts usually induce different results, in attempt to achieve a consensus decision, it is crucial to coordinate the opinions given by different experts. Therefore, our chief goal becomes the appropriate combination of the decisions made by the experts such that the recognition rate can be raised.

Basically, we can apply the voting method or the weighting method to generate a single decision compulsively. Nevertheless, these methods are flawed inherently. Taking voting method for example, it is not always adequate to resort to the majority since the opinion from an authority sometimes is more credible than those from several average experts. On the other hand, the weakness of weighting method is also apparently; experts cannot only be justified according to the domains they specialize in, but also the targets they treat with.

Recently, scholars discovered the importance of decision combination among multiple experts [1]-[3]. T. K. Ho et al [1] tried logistic regression, which has been widely used in the field of statistics, in order to solve alphanumeric character recognition problems. For each class, they used a binary variable Y associated with to indicate an unknown testing sample belonging to the class (Y=1) or not (Y=0). In other words, for each class, the solution space has been partitioned into two subspaces (accepting region and rejecting region). This concept is similar to that of neural networks, which use hidden layer to achieve the partition of the solution space. In case of four classifiers (experts), they found the best recognition rate was improved by 7.8% compared with the best one provided by a single classifier. However, this approach would encounter a serious problem as the number of classes becomes larger: the process of parameter estimation could not converge and therefore the parameters for logistic regression could not be found. S. Huang and C. Y. Suen [2] introduced an approach, called Linear Confidence Accumulation method (LCA) for the combination of classifiers, in a context that each classifier can offer not only class labels but also the corresponding measurement values. They gathered all measurement values for each classifier and calculated the possibility of being a class as a particular measurement value appears. During the online recognition process, for a testing sample X, each classifier calculates the possibilities that X belongs to a class. For each class, take average of the possibilities given by each classifier, and we can obtain the final possibility of the class. Then, the class with the highest probability is selected as the class of X. In fact, LCA can be regarded as a variant of voting method. The LCA collects the voting results in advance, and then take advantage of the statistic data to do character recognition. The weakness of LCA lies in the necessity of analyzing tremendous amount of data to take effect. Moreover, if the features being extracted is not good enough, the more the data being gathered, the lower the recognition rate.

Besides, it is acknowledged that Neural Networks (NN) use supervised learning strategy for weight adjustment. Given a testing sample and a desired output, the weights among hidden-layer neurons can be well adjusted through back propagation. However, some drawbacks of NN have been criticized for a long time, such as tremendous learning time and good initial weights required to achieve good results. Therefore, we are motivated to devise a new scheme for fast weight adjustment. Based on the above observations, we propose a new approach, named MEDC (Multiple Expert Decision Combination), to support a combined decision among multiple experts. It takes advantage of the philosophy underlying neural networks and refines the traditionally weighting method to achieve our goal.

The remainder of this paper is organized as follows. In Section 2, we define the symbols used in this paper. Section 3 introduces our approach for multiple-expert decision combination. Section 4 shows experimental results of our system. Finally, Section 5 summarizes our approach.

2. Symbol Definition

The definition of symbols is similar to that of [2]. e_k means classifier(expert) k where k = 1, …, K, and K is the total number of classifiers. $C_1, \ldots, C_N$ are mutually exclusive and exhaustive sets of patterns. N represents the total number of pattern classes. $\Lambda = \{1, \ldots, N\}$ is a set which consists of all class index numbers. x denotes an input pattern and $e_k(x) = \{m_k^i(x) | \forall i (1 \le i \le N)\}$ means that expert k assigns the input x to each class i with a measurement value $m_k^i(x)$. Then the problem becomes – When K experts give their individual decisions about the identity of a unknown input, how can these individual decisions be combined efficiently to produce a better decision? To formulate this problem, it becomes

$$\left.\begin{array}{l} e_1(x) = m_1^1(x), \ldots, m_1^N(x) \\ e_2(x) = m_2^1(x), \ldots, m_2^N(x) \\ \qquad\qquad \ldots \end{array}\right\}$$

$$\begin{array}{l} e_K(x) = m_K^1(x), \ldots, m_K^N(x) \\ \rightarrow \quad M^1(x), \ldots, M^N(x) \\ \rightarrow \quad E(M^1(x), \ldots, M^N(x)) = j \end{array} \qquad (1)$$

where $M^i(x)$ is the combined measurement value of class i ($1 \le i \le N$), and E is the decision making function of the multiple classifiers which gives x one definitive class j and $j \in \Lambda$.

3. Decision Combination

There are many types of measurement values output by various classifiers such as bitmap distance, crossing counts and junctions. Since these values range between different intervals, they have been normalized into the same interval [0-9999] for the ease of combination. Suppose $m_k^i(x)$ only has contribution to the combined measurement value of the class i, then a general equation to aggregate multiple measurement values is

$$M^i(x) = F(m_1^i(x), \ldots, m_k^i(x), \ldots, m_K^i(x)) \qquad (2)$$

where $M^i(x)$ denotes the combined measurement value of class i ($1 \le i \le N$) and F is the aggregating function which contains parameters $m_k^i(x)$ ($1 \le k \le K$). The most common and simple model of function F performs a weighted and linear summation as

$$M^i(x) = w_1 * m_1^i(x) + \ldots + w_k * m_k^i(x) + \ldots + w_K * m_K^i(x) \qquad (3)$$

where w_k is the weight of classifier k ($1 \le k \le K$). Usually, these weights are adjusted only in the design phase and remained fixed during operation. However, due to inconsistency of measurement values, a constant weight cannot serve its role well. Therefore, a modified aggregation function becomes

$$M^i(x) = w_1^i * m_1^i(x) + \ldots + w_k^i * m_k^i(x) + \ldots + w_K^i * m_K^i(x) \qquad (4)$$

Suppose each sample x contains both $I(x)$ and $\{m_k^i(x) | \forall i, k \ (1 \leq i \leq M \text{ and } 1 \leq k \leq K)\}$, where $I(x)$ is the expected class label of x. The system is illustrated in Fig. 1.

The remainder of this section will discuss the following two issues: (1) the transformation from a measurement value into a normalized measurement value, and (2) weight adjustment method and the decision rule which gives x one definitive class label j.

3.1 Measurement Value

For an input pattern x, expert k assigns the input x to each class i with a measurement value $m_k^i(x)$. In our system, $m_k^i(x)$ represents the matching distance between x and C_i from the viewpoint of expert k. For the ease of combination, all measurement values were normalized to the same interval [0-9999]. For each expert, we gathered a significant amount of measurement values for all possible (x, C_i) pairs, and normalized them to [0-9999] according the similarity between x and C_i. In other words, the more the similarity between x and C_i, the less the measurement value $m_k^i(x)$.

In addition, as expert k assigns measurement value $m_k^i(x)$ to class i, we give a weight w_k^i to this decision according to the credibility of the expert in facing class i. In our system, all weights range in the same interval [0-100] and

$$w_1^i + \ldots + w_j^i + \ldots + w_K^i = 100 (i=1..N). \quad (5)$$

From another point of view, the credibility of expert k's decision that x belongs to class i is embodied by weight w_k^i. Weight w_k^i also indicates the percentage of the final decision contributed by expert k. Then, the combined measured value to class i, $M^i(x)$, can be obtained from equation (4).

The next step is to derive a final decision from the set of $M^i(x)$, where i=1 to N. The decision rule applied in our approach is

$$E(x) = \text{argmin}_{1 \leq i \leq N} \{ M^i(x) \} \qquad (6)$$

In other words, x will be classified to the class with least combined measurement value.

3.2 Weight Adjustment

The weights are adjusted off-line, using significant amount of training samples gathered beforehand. To adjust weights effectively, we can take advantage of the discrepancy between combined decision and individual experts. We can either increase the weight of the expert whose decision is better than the combined decision, or decrease the weight of the expert whose decision is worse than the combined decision. This concept is illustrated by an example with 4 experts and 8 classes as shown in Table I.

In Table I, The expected class for the input pattern x whose is C_5. From the viewpoint of expert e_1, C_7 is the one that is most similar to x; C_3 is in the second place, and the expected class C_5 is in the third place and so on. From the viewpoint of combined rank, C_5 is in the third place. The combined rank for each class is obtained from the weighted sum of the ranks provided by each expert. Initially, all weights are equally assigned. We can find that the combined rank of class C_1 and class C_3 are better than that of class C_5. Since the final decision is according to the combined rank, we can regard C_1 and C_3 as the obstructers that hinder C_5 from being the first candidate. In order to promote the combined rank of C_5, we can either increase the weight of the expert whose decision is better than the combined decision, or decrease the weight of the expert whose decision is worse than the combined decision.

On one hand, to increase the weight of the expert whose decision is better than the combined decision, the following two operations can be applied:

OP1: Considering the expected class (C_5), increase the weight of the expert whose decision is better than the combined decision rank. For instance, since *OP1* can be applied to expert e_2, the weight of expert e_2 is increased by ¢. That is

$$w_2^5 = w_2^5 + ¢ \qquad (7)$$

OP2: Considering the obstructive class (eg. C_1), increase the weight of the expert whose decision is better than the combined decision. For instance, since *OP2* can be applied to expert e_3, the weight of expert e_3 is increased by ¢. That is

$$w_3^1 = w_3^1 + \text{¢} \qquad (8)$$

On the other hand, to decrease the weight of the expert whose decision is worse than the combined decision, the following two operations can be applied:

OP3: Considering the expected class (C_5), decrease the weight of the expert whose decision is worse than the combined decision rank. For instance, since *OP3* can be applied to expert e_1, the weight of expert e_2 is decreased by ¢. That is

$$w_1^5 = w_1^5 \; J \; \text{¢} \qquad (9)$$

OP4: Considering the obstructive class (eg. C_1), decrease the weight of the expert whose decision is worse than the combined decision. For instance, since *OP4* can be applied to expert e_4, the weight of expert e_4 is decreased by ¢. That is

$$w_4^1 = w_4^1 \; J \; \text{¢} \qquad (10)$$

In fact, operation *OP1* and *OP3* can achieve the same goal since $w_1^i + \ldots + w_4^i = 100$ (i = 1..8). In other words, considering class C_i, we have to decrease the weights of other experts while increasing the weight of an expert, and vice versa. On the other hand, the effect of operation *OP2* and *OP4* are similar. In sum, only two weight adjustment methods are indispensable.

4.Experimental Results

Our system is implemented on a Pentium II 400 personal computer and the programming language is Visual C++. Table II shows the experimental results of recognizing 1,000 Black-font and 1,000 Kai-font Chinese characters, using 1,000 Ming-font Chinese characters as templates. Eight classifiers are used in our experiments. In case of using Black-font characters as testing samples, the recognition rate is 96.2 percent and the average rank of expected class is 1.12 for the best single classifier. Through the combination of classifiers, the recognition rate is initially improved to 98.4 percent and further improved to 99.3 percent after weight adjustment. The CPU time used for weight adjustment is 0.431 sec. In other words, through MEDC, the recognition rate was improved by 3.22% compared with the best one provided by a single classifier. Similarly, in case of using Kai-font characters as testing samples, the recognition rate is improved by 9.99%. Since, compared with Kai-font, Black-font is more similar to Ming-font, the result of using Black-font characters as testing samples is better than that of using Kai-font characters.

5.Conclusions

In this paper, we demonstrated a classification system, called MEDC, based on multiple expert decision combination. It takes advantage of the philosophy underlying neural networks and refines the traditional weighting methods to achieve our goal. Experimental results show that, through MEDC, the recognition rate was improved by 9.99% compared with the best one provided by a single classifier. In our future works, MEDC will be applied to other areas in expert systems and pattern recognition.

References

[1] T. K. Ho, J. J. Hull and S. N. Srihari, “Decision combination in multiple classifier system,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no.1, pp. 66-75, 1994.

[2] Y. S. Huang and C. Y. Suen, “Combination of multiple classifiers with measurement values,” in *Proc. Int. Conf. Document Analysis and Recognition*, pp. 598-601, 1993.

[3] A. F. R. Rahman and M. C. Fairhurst, “Introducing new multiple expert decision combination topologies: a case study using recognition of handwritten characters,” in *Proc. Int. Conf. Document Analysis and Recognition*, pp. 886-891, 1997.

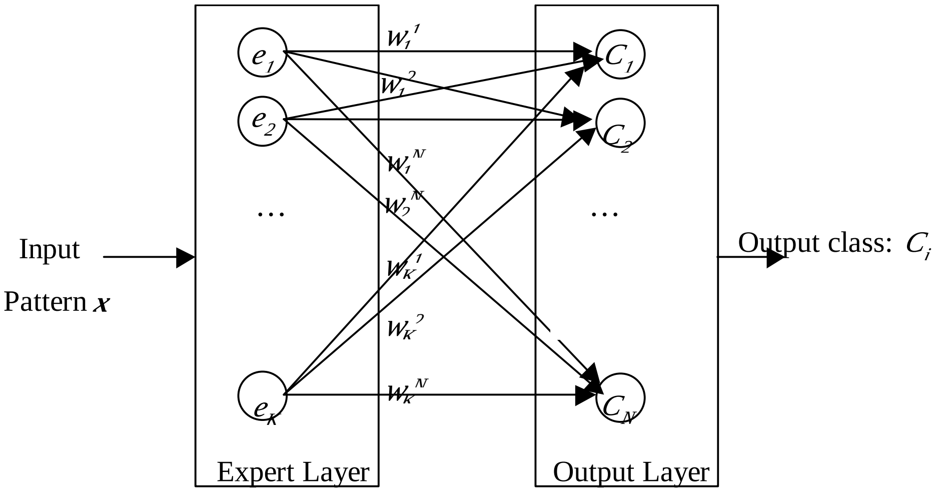

Figure 1. The multiple-expert decision combination system.

Table I

An example of 4 experts and 8 classes.

Rank	1	2	3	4	5	6	7	8
Combined Rank	C_1	C_3	$\boldsymbol{C_5}$	C_2	C_4	C_6	C_7	C_8
Expert e_1	C_7	C_3	C_2	$\boldsymbol{C_5}$	C_4	C_6	C_1	C_8
Expert e_2	$\boldsymbol{C_5}$	C_3	C_1	C_2	C_4	C_6	C_7	C_8
Expert e_3	C_4	C_3	C_6	C_2	C_1	$\boldsymbol{C_5}$	C_7	C_8
Expert e_4	C_1	$\boldsymbol{C_5}$	C_6	C_2	C_8	C_3	C_7	C_4

Table II

Results of recognizing 1,000 Black-font and 1,000 Kai-font Chinese characters (using 1,000 Ming-font Chinese characters as templates)

	Black Font		Kai Font	
	Recognition Rate	Average Rank	Recognition Rate	Average Rank
Best Single Classifier	0.962	1.12	0.891	1.65
Initial Weight	0.984	1.02	0.881	1.263
After Weight Adjustment	0.993	1.007	0.98	1.053
CPU time used for Weight Adjustment (sec.)	0.431		0.437	
Percentage of Improvement	3.22		9.99	