Confidence Combination Methods in Multi-expert Systems

Yingquan Wu, K. Ianakiev, and V. Govindaraju

Center for Excellence in Document Analysis and Recognition (CEDAR)
Department of Computer Science & Engineering
University at Buffalo, Amherst, NY 14228, USA
{ianakiev,govind}@cedar.buffalo.edu
Fax: (716) 645-6176

Abstract. In the proposed paper, we investigate the combination of the multi-expert system in which each expert outputs a class label as well as a corresponding confidence measure. We create a special confidence measurement which is common for all experts and use it as a basis for the combination. We develop three combination methods. The first method is theoretically optimal but requires very large representative training data and storage memory for look-up table. It is actually impractical. The second method is suboptimal and reduces greatly the required training data and memory space. The last method is a simplified version of the second and needs the least training data and memory space. All three methods demand no mutual independence of the experts, thus should be useful in many applications.

Keywords: Expert, classifier, combination methods, OCR, confidences, Bayes rule

1 Introduction

In the area of pattern recognition, practical applications require highly reliable classification which may be difficult for a single algorithm to achieve. Since there are a number of classification algorithms in the literature, based on different theories and methodologies, a combination of these can be used to improve the classification performance by taking advantages of their strengths and avoiding their weaknesses. The task is quite challenging because the decisions of the individual experts are conflicting.

The idea of combining the decisions of multiple experts has been explored by many researchers [1]–[15]. In general, based on the output information, there are three types of experts: Type I that outputs a unique class label indicating the most probable class to which the input pattern belongs; Type II that outputs a ranked list of part or all of class labels such that the higher a class label is in the list, the more probable it is that the input pattern belongs to the corresponding class; Type III that assigns to each class label a measurement value which indicates the degree by which the corresponding class pertains to the input pattern.

F.J. Ferri et al. (Eds.): SSPR&SPR 2000, LNCS 1876, pp. 641–649, 2000.

Combining Type III experts is the most challenging, because of the potentially many possible combinations of measurement values and the complicated relations between these values and the experts performance. The problem is even further complicated by the lack of standard measurements and in this sense, the measurement values of different experts are usually not compatible.

Previous studies have developed many different approaches for expert combination. For experts of Type I for which only labels are available, voting algorithms are used ([1], [2], [3]). Label rankings are used with experts of Type II ([4], [5], [3]). In case of Type III experts with measurement values interpreted as posteriori probabilities, a Bayesian technique is often applied for experts assemblies ([6], [7]). If the expert output is interpreted as fuzzy membership or evidence values, fuzzy rules ([8],[9]) and Dempster-Shafer approach ([6],[10], [11]) are used. Also there are cases of expert combination, where the output of the expert is used as a new feature and a new expert (neural network) is built to perform the combination ([12], [13], [14]).

In this paper, we focus on a simplified version of Type III multi-expert combination. Each expert uses its own representation, i.e. measurements extracted from the input pattern are unique to each expert. We create a new accuracy measurement scale, uniform for all experts and use it as a basis for the expert combination. We develop an optimal combination scheme which requires extremely large amount of training data as well as memory space. So we introduce an empirical scheme to approximate the optimal scheme so that the requirements on the ammount of training data and memory is practical. We first develop the accuracy measure which is common for all experts. Next, we characterize each expert with a family of accuracy maps. Next, we build accuracy combination maps with a special synthetic function. Finally, we construct synthetic accuracy maps for the combined confidences. We also propose a simplified combination scheme which requires less training data while sacrificing some accuracy. We finally discuss the optimal rejection threshold for the final recognition decision.

In Section II, we state the problem formulation. In Section III, we introduce the accuracy approximation method. We describe the optimal combination rule in Section IV and an empirical combination rule in Section V. We derive the optimal rejection threshold in Section VI. We give simulation results in Section VII and finally draw conclusions in Section VIII.

2 Problem Formulation

Many classifiers are able to supply confidence information from the measurement level. Bayes' classifier supplies the a posteriori probabilities as confidence measurements. Various distance related classifiers use the distance between a test pattern and template/prototype patterns as confidence measurement.

In this paper, we assume that the given experts only supply the top choice and the corresponding confidence. Specifically, let $E^{(n)}$ represent expert (classifier) n, where $n = 1, 2, \ldots, N$, and N is the total number of experts. $\Lambda = \{C^{(1)}, C^{(2)}, \ldots, C^{(M)}\}$, are mutually exclusive and exhaustive set of class la-

bels. $E^{(n)}(\mathbf{x}) = (E_1^{(n)}(\mathbf{x}), E_2^{(n)}(\mathbf{x}))$ means that expert n assigns the unknown pattern $\mathbf{x}$ to class label $E_1^{(n)}(\mathbf{x}) \in \Lambda$ with confidence $E_2^{(n)}(\mathbf{x}) \in [0, 1]$.

Our task is to approximate the conditional accuracy distribution function for a multi-expert system $P(E_1^{(i)} |\ (E_1^{(1)}, E_2^{(1)}), (E_1^{(2)}, E_2^{(2)}), \ldots, (E_1^{(N)}, E_2^{(N)}))$, $i = 1, 2, \ldots, N$ by using the given training samples.

3 Accuracy Approximation

Recognizers usually differ in their confidence measures. A judicious combination of these measures can be made only when they are on the same scale. We will develop a way to transform confidence measures from different recognizers to a "special" confidence, we call accuracy. For a given set of "large enough" as well as "properly representative" training samples, the accuracy measure can be obtained by transforming the confidence using sufficient training samples. Usually, the accuracy is an increasing function of confidence.

Let us assume that the training set is adequately representative. Now let us discuss the accuracy approximation by utilizing the training samples. We assume that the confidence measures of all classifiers are continuous, that is, the confidence values can be any point in [0, 1]. There are many methods that can implement this transform. Here we introduce a simple and efficient method.

Let L patterns be classified to a certain class label by an expert and have confidence between $[a, b)$. Let t out of L patterns be correctly classified, then we can assign the approximate accuracy over $[a, b)$ as $\tilde{\mu}(r) = t/L,\ \forall r \in [a, b)$. For a given error bound ε, we claim that L has statistical sense if the probability that the approximate accuracy $\tilde{\mu}(r), r \in [a, b)$ is within the error ε from any true accuracy value $\mu(r), \forall r \in [a, b)$ is greater than $1 - \varepsilon$, that is,

$$Pr(\max_{r \in [a,b)} \{|\tilde{\mu}(r) - \mu(r)| < \varepsilon\}) > 1 - \varepsilon. \tag{1}$$

According to (1), the difference of two accuracy values between two adjacent representative intervals $[a, b)$ and $[b, c)$ satisfies

$$Pr(|\tilde{\mu}(r_2) - \tilde{\mu}(r_1)| < 2\varepsilon) > 1 - 2\varepsilon, \quad \forall r_1 \in [a, b),\ \forall r_2 \in [b, c). \tag{2}$$

At the same time, the accuracy value is within [0, 1]. So the number of representative intervals over $[0, 1]$ must be greater than $1/(2\varepsilon)$. Therefore, the number of training samples to approximate accuracy is of the order of $O(L/(2\varepsilon))$ for a class label. Since there are M labels, totally $O(ML/(2\varepsilon))$ samples are required to capture the accuracy characteristic of an expert with error tolerance within ε.

In the real implementation, for a given ε we are not able to estimate L according to (1). Instead, we estimate L by making the difference of two accuracy values between two adjacent representative intervals $[a, b)$ and $[b, c)$ less than 2ε, that is,

$$0 \leq \tilde{\mu}(r_2) - \tilde{\mu}(r_1) < 2\varepsilon, \quad \forall r_1 \in [a, b),\ \forall r_2 \in [b, c). \tag{3}$$

4 Optimal Combination Rule

Let us assume we are given sufficient number of representative training samples so that we are able to generate the conditional accuracy distribution function for a multi-expert system, $P(E_1^{(i)} | (E_1^{(1)}, E_2^{(1)}), (E_1^{(2)}, E_2^{(2)}), \ldots, (E_1^{(N)}, E_2^{(N)}))$, $i = 1, 2, \ldots, N$. This "Behavior-Knowledge Space" scheme produces the optimal combination performance [13].

However, such approach needs large number of samples. Let us find the lower bound on the number of samples for the above accuracy approximation with error bound ε. There are $O(M/(2\varepsilon))$ choices for each pair $(E_1^{(i)}, E_2^{(i)})$, $i = 1, 2, \ldots, N$ and thus $[M/(2\varepsilon)]^N$ choices for $\{(E_1^{(1)}, E_2^{(1)}), (E_1^{(2)}, E_2^{(2)}), \ldots, (E_1^{(N)}, E_2^{(N)})\}$. For each choice, at least L samples are required. Therefore, a total number of $O(L[M/(2\varepsilon)]^N)$ samples are necessary to build the above "Behavior-Knowledge Space". Also, the memory of $O(N[M/(2\varepsilon)]^N)$ are required to build the look-up table of the joint accuracy distribution.

5 Empirical Combination Rules

5.1 Accuracy Combination Functions

The accuracy combination function is of the form $F(a_1, a_2, \ldots, a_n)$, where $a_i \in [0, 1], 1 \leq i \leq n$ are n accuracy variables. $F(\cdot)$ is supposed to be symmetric and in [0, 1]. Moreover, $F(\cdot)$ must satisfy the following two special properties.

$$F(a_1, a_2, \ldots, a_n) = 1, \text{ if } a_k = 1, \exists k; \tag{4}$$

$$F(a_1, a_2, \ldots, a_n) = 0, \text{ if } a_k = 0, \exists k. \tag{5}$$

The justification of (4) and (5) is obvious. In fact, it is never the case that $a_i = 1$ at the same time as $a_j = 0$ (theoretically). Therefore, in this case the function $F(\cdot)$ is supposed to be non-existent.
A family of functions satisfying the required conditions are:

$$F(a_1, a_2, \ldots, a_n) = [h^{-1}(\frac{1}{n}\sum_{i=1}^{n} h(a_i^{\lambda}))]^{1/\lambda}, \quad \lambda > 0. \tag{6}$$

where λ is a parameter and $h(r)$, $r \in [0, 1]$ is a strictly ascending function satisfying

$$h(0) = -\infty; \tag{7}$$

$$h(1) = \infty. \tag{8}$$

Here, we consider ∞ as an existent number. Four simple examples of $h(\cdot)$ are listed as follows:

$$h(r) = \tan(\pi(r - 1/2)), \quad r \in [0, 1]; \tag{9}$$

$$h(r) = (1/2 - |r - 1/2|)^{-1}(r - 1/2), \quad r \in [0, 1]; \tag{10}$$

$$h(r) = (r - r^2)^{-1}(r - 1/2), \quad r \in [0, 1]; \tag{11}$$

$$h(r) = (r - r^2)^{-1/2}(r - 1/2), \quad r \in [0, 1]. \tag{12}$$

5.2 Combination Scheme

For each classification distribution of the multi-expert system $(C_1, C_2, \ldots, C_N)$, we build an accuracy map, called characteristic accuracy map for each expert. We note that there are M^N possibilities. So we build M^N characteristic accuracy maps in total for each expert. In practice, we might not have all the M^N possibilities, since some of permutations do not exist. The required number of training samples with error bound ε is $O(M^N L/(2\varepsilon))$. We note the required training samples is just linear in the number required to approximate an accuracy map, thus greatly reducing the required number of training samples.

For a given distribution $(C_1, C_2, \ldots, C_N)$, we denote a set of training samples by χ such that

$$(E_1^{(1)}(\mathbf{x}), E_1^{(2)}(\mathbf{x}), \ldots, E_1^{(N)}(\mathbf{x})) = (C_1, C_2, \ldots, C_N), \quad \forall \mathbf{x} \in \chi. \tag{13}$$

Let Γ denote the expert indices set such that $C_i = C, \forall i \in \Gamma$ and $|\Gamma| \geq 2$. Let $\tilde{E}_2^{(i)}$, $i \in \Gamma$ denote the characteristic accuracy maps constructed from χ and $E_1^{(i)}(\mathbf{x}) = C$, $\forall i \in \Gamma$, $\forall \mathbf{x} \in \chi$. We define the synthetic accuracy as

$$E_2^{(\Gamma)}(\mathbf{x}) = F(\tilde{E}_2^{(i)}(\mathbf{x}) : i \in \Gamma). \tag{14}$$

When the combination scheme is given, we can easily get the combination accuracy $\tilde{E}_2^{(\Gamma)}$ from the synthetical confidence $E_2^{(\Gamma)}$ using the data set χ. The generated accuracy map is supposed to be an ascending function of the combination confidence.

Now let us discuss the maximum memory required for the look-up table for all accuracy maps. For simplicity, we just consider the number of accuracy maps. The number of synthetic accuracy maps which combines exactly n accuracies are $M\binom{N}{n}(M-1)^{N-n}$. The total number of synthetic accuracy maps is $\sum_{n=2}^{N} M\binom{N}{n}(M-1)^{N-n} = M(M^N - N(M-1)^{N-1} - (M-1)^N)$. The maximum number of accuracy maps, including characteristic and synthetic maps, are $M(M^N - N(M-1)^{N-1} - (M-1)^N) + NM^N$.

5.3 Simplified Combination Scheme

In the original scheme, we need to build $O(M^N)$ accuracy maps for each expert. Usually, M is a large number, e.g., $M = 10$ in numerical recognition; $M = 10+26\times 2 = 62$ in character recognition. So M^N can be a very large number even for $N = 2$. Thus this method still needs quite a large amount of training data. So instead of collecting training data for each specific distribution $(C_1, C_2, \ldots, C_N)$ in the original scheme, we can collect training data for each case such that $C_i = C$, $\forall i \in \Gamma$, where $C \in \Lambda$, and Γ is a set of expert indices such that $|\Gamma| \geq 2$. This method requires smaller set of training data, however this is gained by sacrificing the performance. This method performance is inferior to the original scheme when given sufficient representative training data.

The synthetic accuracy map construction procedure is same as the original scheme. For a special expert n, expert index set Γ including index n has 2^{N-1} choices and class label C has M choices. Thus we need to build $2^{N-1}M$ characteristic accuracy maps for each expert. Therefore $O(2^{N-1}ML/(2\varepsilon))$ limit samples are required. In return, we curtail the required number of training samples by $O((M/2)^{N-1})$ times. When $M = 2$, both schemes become identical.

Now let us discuss the maximum number of accuracy maps. The number of synthetic accuracy maps which combines exactly n accuracies are $M\binom{N}{n}$. The total number of synthetic accuracy maps is $\sum_{n=2}^{N} M\binom{N}{n} = M(2^N - 1 - N)$. The maximum number of accuracy maps, including characteristic and synthetic maps, are $M(2^N - 1 - N + N2^{N-1})$. In comparision with the original combination scheme, we also reduce the necessary memory by $O((M/2)^{N-1})$ times.

6 Optimal Rejection Threshold

In the final stage of making the recognition decision, we have to make one of two decisions: acceptance or rejection. There is a cost associated with both error as well as rejection. Trade-offs between the rejection and error ratio must be made. We follow the optimization objective as in [15]

$$R_{obj} = \min\{R_{err} + \alpha R_{rej}\}. \tag{15}$$

where R_{err} and R_{rej} are error ratio and rejection ratio, respectively, and $0 \leq \alpha \leq 1$ is a deterministic parameter.

A natural way to determine the recognition class is to choose the class which has the maximum accuracy upon the proposed scheme. However, we need to decide to discard or accept the recognition class according to the recognition accuracy.

Let us determine the accuracy threshold which minimizes the objective defined in (15). In fact, we can explcitly express both items R_{err} and R_{rej} as functions of a threshold θ as follows:

$$R_{err} = \int_{r=\theta}^{1} (1 - r)dr = 1/2 - \theta + \theta^2/2. \tag{16}$$

$$R_{rej} = \int_{0}^{\theta} 1dr = \theta. \tag{17}$$

Hence, we have

$$R_{obj} = \min_{\theta \in [0,1]} \{1/2 - \theta + \theta^2/2 + \alpha\theta\} \tag{18}$$

Taking the derivative of both sides with respect to θ, we obtain

$$0 = -1 + \theta + \alpha \tag{19}$$

Thus the optimal accuracy threshold θ is given by:

$$\theta = 1 - \alpha. \tag{20}$$

That is, when recognition accuracy is less than the decision threshold $\theta = 1 - \alpha$ we reject the result, otherwise we accept it.

When $\alpha = 1$, rejecting an unknown pattern is equivalent to misclassification, as we would like to accept all recognition. This verifies the optimal threshold $\theta = 0$. When $\alpha = 0$, the objective is to minimize the error rate alone, so we would accept an unknown pattern only if it has recognition accuracy 1, that is, no error is made (theoretically). This again verifies the optimal threshold of $\theta = 1$.

7 Experimental Results

The training set used in the construction of the accuracy approximation and the testing set were created using digit samples extracted from the US mail stream. There are two reasons why we use our own database. First, the recognizers used in the combinations schemes achieve almost 100% correct rate on databases available publicly, such as NIST. Second, all classes are equally represented in the training set which is not the case with other databases. The training set contains 120,000 digit samples, and the testing set contains 30,000 digit smaples.

Table 1. Performances of binpoly and gradient experts

	binpoly			gradient		
α	err(%)	rej(%)	opt	err(%)	rej(%)	opt
1/5	4.49	31.61	54.04	1.92	12.15	21.73
1/10	0.19	82.64	84.52	1.41	17.30	31.35
1/15	0.19	82.64	85.46	1.03	22.27	37.77
1/20	0.19	82.64	86.46	0.82	26.37	42.86

We used the simplified method exactly as it is described in section V.3 and the thresholds given in the previous section. Table 1 shows the performance of the "binpoly" expert [16] and the performance of the "gradient" expert [17]. The "binpoly" expert is a polynomial discriminant algorithm trained to extract a relative weighting for each feature in each class. The "gradient" expert encodes local contour variations of the character image into a binary feature vector.

Table 2. Performance of binpoly-gradient combination

α	err(%)	rej(%)	opt
1/5	1.44	9.34	16.54
1/10	0.97	12.88	22.63
1/15	0.76	15.42	26.88
1/20	0.69	17.20	30.92

Table 3. Performance of kp and gsc experts

	kp			gsc		
α	err(%)	rej(%)	opt	err(%)	rej(%)	opt
1/5	0.89	10.96	15.44	1.16	4.99	10.80
1/10	0.78	12.00	19.79	1.00	6.19	16.14
1/15	0.67	13.73	23.84	0.90	7.57	21.14
1/20	0.47	16.25	25.61	0.85	8.58	25.58

Table 2 shows the results of the combination of these two experts. As we can see, for all values of α, the combination method got significant improvement of the objective function - 23.93% for $\alpha = 0.2$; 27.75% for $\alpha = 0.1$; 28.86% for $\alpha = 0.067$; and 27.85% for $\alpha = 0.05$ compared to the values of the objective function for the better expert.
Table 3 describes the performance of the "kp" expert (unpublished) and the performance of "gsc" expert [18]. Table 4 shows the results of the combination of these two experts. The "kp" expert combines the merits of "binpoly" expert and "gradient" expert. "GSC" expert extracts features based on gradient, struactural, and concavity. As we can see, these are much more accurate experts, nevertheless for all values of α, the combination method improve the objective function - 5.56.% for $\alpha = 0.2$; 6.82% for $\alpha = 0.1$; 13.71% for $\alpha = 0.067$; and 17.20% for $\alpha = 0.05$ compared to the values of the objective function for the better expert.

Table 4. Performance of kp-gsc expert combination

α	err(%)	rej(%)	opt
1/5	1.10	4.73	10.22
1/10	0.82	6.85	15.05
1/15	0.66	8.41	18.25
1/20	0.60	9.23	21.17

8 Conclusion

We have investigated the simplified version of type 3 multi-expert systems in which each expert outputs a class label and a corresponding confidence. We have developed a general theoretical framework for optimal posterior-probability based combination scheme and have shown that it needs a huge representative training set as well as large memory. This is impractical. We have therefore developed an empirical approach to approximate the joint accuracy distribution function. In this approach, we develop a special measurement, accuracy, which is applicable to all experts. We characterize each expert with a class of accuracy maps. We also develop a family of special combination functions. Finally, we have discussed the optimal accuracy threshold for the recognition decision.

This approach doesn't require mutual independence of experts. In fact, [13] is just a special case of our approach. However, all desirable properties exist from the statistical point of view. A "large enough" and "well represented" training sample set must be available. If only few samples are collected randomly and carelessly, the desired properties of this method cannot be guaranteed [13].

References

1. F. Kimura, M. Shridhar, "Handwritten numerical recognition based on multiple algorithms", *Pattern Recognition* vol. 24, no. 10, pp. 969–983, 1991.
2. C. Y. Suen, R. Legault, C. Nodel, M. Cheriet, L. Lam, "Building a new generation of handwriting recognition systems", *Patterns Recognition Letters*. vol. 16, no. 1, pp. 66–75, 1994.
3. K. Al-ghoneim and B. V. V. Kumar, "Unified decision combination framework", *Pattern Recognition*, vol. 31, no. 12, pp. 2077–2089, 1998.
4. S. C. Bagui and N. R. Pal, "A multi-stage generation of the rank nearest neighbour classification rule", *Pattern Recognition Letters*, vol. 16, no. 6, pp. 601–614, 1995.
5. T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems", *IEEE Trans. Patterns Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1995.
6. L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their application to the handwritten numerical recognition", *IEEE Trans. on System, Man, and Cybernetics*, vol. SHC-22, no. 3, pp. 418–435, 1992.
7. Y. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Interference*, San Mateo, CA: Morgan Kaufmann Publishers, 1988.
8. S. B. Cho, J. H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification", *IEEE Trans. Systems, Man, and Cybernetics*, vol. 25, no. 2, pp. 380–384, 1995.
9. S. B. Cho, J. H. Kim, "Multiple network fusion using fuzzy logic", *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 497–501, 1995.
10. G. Rogaua, "Combining the results of several neural network classifiers", *Neural Networks*, vol. 7, no. 5, pp. 777–781, 1994.
11. M. D. Meteish, P. Yao, T. Stirtzinger, "A study on the use of believe functions for medical expert systems", *J, Appl. Statistics*, vol. 18, no. 1, 99. 155–174, 1991.
12. D. H. Wolpert, "Stacked Generalization", *Neural Networks*, vol. 5, no. 2, pp. 241–260, 1992.
13. T. S. Huang, and C. Y. Suen, "A method of Combining multiple experts for the recognition of unconstrained handwritten numerals", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, 1995.
14. J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers", *Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
15. L. Lam and C. Y. Suen, "Optimal combinations of pattern classifiers", *Pattern Recognition Letters*, vol 16, pp. 945–954, 1995.
16. U. Srinivasan, "Polynomial discriminant method for Handwritten digital recognition", Tech. Rep., SUNY Buffalo, December 14, 1989.
17. G. Srikantan, "Image sampling rate and image pattern recognition", Dissertation, SUNY Buffalo, 1994.
18. S. W. Lam, G. Srikantan, and S. N. Srihari, "Gradient-based contour encoding for character recognition", *Pattern Recognition*, vol. 29, no. 7, pp. 1147–1160, 1996.