Neural Networks 63 (2015) 87–93

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Performance improvement of classifier fusion for batch samples based on upper integral

Hui-Min Feng, Xi-Zhao Wang*

Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding 071002, China

ARTICLE INFO

Article history:
Received 2 March 2014
Received in revised form 12 October 2014
Accepted 14 November 2014
Available online 28 November 2014

Keywords:
Extreme learning machine
Upper integral
Fuzzy measure
Fuzzy integral
Multiple classifier fusion

ABSTRACT

The generalization ability of ELM can be improved by fusing a number of individual ELMs. This paper proposes a new scheme of fusing ELMs based on upper integrals, which differs from all the existing fuzzy integral models of classifier fusion. The new scheme uses the upper integral to reasonably assign tested samples to different ELMs for maximizing the classification efficiency. By solving an optimization problem of upper integrals, we obtain the proportions of assigning samples to different ELMs and their combinations. The definition of upper integral guarantees such a conclusion that the classification accuracy of the fused ELM is not less than that of any individual ELM theoretically. Numerical simulations demonstrate that most existing fusion methodologies such as Bagging and Boosting can be improved by our upper integral model.

1. Introduction

Huang, Zhu, and Siew (2004, 2006) proposed a new learning algorithm for single-hidden layer feedforward networks (SLFNs) called Extreme Learning Machine (ELM) which overcomes the problems caused by gradient descent based algorithms such as Back propagation applied in artificial neural networks. ELM can significantly reduce the amount of time needed to train a neural network and preserve the universal approximation ability (Huang, Chen, & Siew, 2006). It randomly chooses the input weights and hidden node biases, and analytically determines the output weights of SLFN. It has much better generalization performance with much faster learning speed (Huang et al., 2006). It automatically determines all the network parameters analytically, which avoids trivial human intervention and makes it efficient in online and realtime applications (Huang et al., 2006; Lan, Soh, & Huang, 2009). ELM has several advantages such as ease of use, faster learning speed, higher generalization performance, suitable for many nonlinear activation function and kernel functions (Liu, He, & Shi, 2008; Wang, Chen, & Feng, 2011).

To achieve good generalization performance, ELM minimizes training error on the entire training data set, therefore it might suffer from overfitting as the learning model will approximate all training samples well (Liu & Wang, 2010). Hansen and Salamon (1990) have showed that the generalization ability of a neural network system can be significantly improved through ensembling a number of neural networks. Combining multiple classifiers to solve a given classification problem is an efficient approach to improve the performance of classification and avoid overfitting (Jain, Duin, & Mao, 2000).

When outputs of a base classifier are real-valued vectors (most often posterior probabilities or possibilities (Kuncheva, 2003), sometimes evidences), a fusion operator such as maximum/minimum, median, average, weighted average, ordered weighted average, Dempster–Shafer approach or fuzzy integral, can be selected to aggregate the outputs from all individual base classifiers (Kuncheva, 2003; Schmitt, Bombardier, & Wendling, 2008; Zhai, Xu, & Li, 2013; Zhai, Xu, & Wang, 2012). The fusion based on maximum/minimum, median or average is suitable for the case that in a combination the importance of base classifier is identical (Kuncheva, 2003; Verikas, Lipnickas, Malmqvist, Bacauskiene, & Gelzinis, 1999). If the importance of a base classifier is different from another, weighted average and ordered weighted average can be chosen (Kuncheva, 2003; Yager, 1988). The importance of a single classifier is emphasized in weighted average while the magnitude of output from a base classifier is particularly considered in ordered weighted average (Kuncheva, 2003; Yager, 1988). But the two methods are under an assumption that interaction does not exist among the individual classifiers. However, this assumption may not be true in many real problems. If the interaction is involved, the fuzzy integral (Schmitt et al., 2008; Wang et al., 2011) or Dempster–Shafer approach (Shafer, 1976) is considered as one of the most appropriate choices. Fuzzy integrals are more

* Corresponding author.
E-mail address: xizhaowang@ieee.org (X.-Z. Wang).

http://dx.doi.org/10.1016/j.neunet.2014.11.004
0893-6080/

computationally efficient than a strict Dempster–Shafer approach (Keller, Gader, Tahani, Chiang, & Mohamed, 1994). The fuzzy integral as a fusion tool, in which the non-additive measure can clearly express the interaction among classifiers and the importance of each individual classifier, has its particular advantages. Additionally the average, weighted average and ordered weighted average can be regarded as special cases of fuzzy integrals. For a tested sample, each base classifier outputs a vector in which the ith component is the degree of the sample belonging the ith class. The fuzzy integral integrates these degrees with respect to a fuzzy measure for each class. One difficulty of applying fuzzy integrals in classifier fusion is how to determine the fuzzy measures. The training process of fuzzy integral fusion method contains training base classifiers and learning the fuzzy measure from training samples. From references one can find a number of methods to determine fuzzy measures such as linear programming, quadratic programming (Yeung, Wang, & Tsang, 2004), genetic algorithm (Yang, Wang, Heng, & Leung, 2008), neural network (Wang & Wang, 1997), and pseudo-gradient (Wang, Leung, & Klir, 2005).

This paper proposes a new approach to multiple classifier fusion based on the upper integral which is a type of fuzzy integrals proposed by Wang, Li, and Leung (2008). Motivated by the definition of upper integrals which can be considered as a mechanism of maximizing potential efficiency of classifier combination, the new approach is devoted to improve the classification performance of a fusion operator based on upper integrals. It is worth noting that, in our approach, the upper integral itself is not considered as a tool of classifier-fusion but it is considered as a tool to improve any existing classifier-fusion operator. In other words, our approach (in which the upper integral is no longer a fusion operator) differs from all existing fuzzy integral based fusion schemes (which consider the fuzzy integrals as fusion operators). Specifically, given a group of individual classifiers trained from a set of samples and a fusion operator, we regard the classification accuracies of individual classifiers and their combinations as the efficiency measure, which avoids almost the difficulty of determining fuzzy measures. The upper integral plays a role of assigning suitable proportion of samples to different individual classifiers and their combinations to obtain maximum the classification efficiency. It computes how many samples will be allocated to some of individual classifiers and their combinations by solving an optimization problem derived from the upper integral. This implies a proportion of sample-allocation for a given set of samples. Based on this proportion, some oracles are used to determine which samples will be allocated to those individual classifiers and their combinations. Given a sample, the oracle of a combination of classifiers first predicts the possibility with which the combination can correctly classify the sample. Then the sample is allocated to the combination with maximum possibility. When the number of samples allocated to a combination attains the proportion, the allocation to this combination stops, and the allocations to other combinations continue until all samples are allocated. After the allocation, those classifiers perform the classification of the set of samples, which is our final classification result.

The rest of this paper is arranged as follows. In Section 2, the existing multiple classifier fusion schemes are reviewed. Section 3 is devoted to the efficiency measures, fuzzy integrals and upper integrals. Our proposed new fusion scheme based on the upper integral is given in Section 4. Section 5 presents a number of numerical experiments to verify advantages of the new approach, and finally Section 6 concludes this paper.

2. Multiple classifier fusion based on fuzzy integrals

Suppose that $X = \{x_1, x_2, \ldots, x_n\}$ is a set of classifiers. The output of classifier x_i is a c-dimensional nonnegative vector $[d_{j,1}, d_{j,2}, \ldots, d_{j,c}]$ where c is the number of classes. Without loss of generality, let $d_{j,i} \in [0, 1]$ denote the support from classifier x_j to the hypothesis that the sample submitted for classification comes from the ith class C_i for $j = 1, 2, \ldots, n$, $i = 1, 2, \ldots, c$. The larger the support, the more likely the class label C_i. All outputs of classifiers for a particular sample can be organized in a matrix

$$DP = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,i} & \cdots & d_{1,c} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,i} & \cdots & d_{2,c} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,i} & \cdots & d_{n,c} \end{bmatrix}.$$

Each column of DP matrix can be regarded as a function defined on the classifier set X, $f_i : X \to [0, 1]$, $f_i(x_j) = d_{j,i}$, $i = 1, 2, \ldots, c$, $j = 1, 2, \ldots, n$. For each class C_i, we need to determine a nonnegative set function μ_i on the power set $P(X)$ of X. μ_i can represent not only the importance of individual classifiers but also the interaction among classifiers towards samples from C_i class. Set functions have some special cases.

Definition 1 (*Wang et al., 2008*)**.** Let X be a nonempty and finite set and $P(X)$ be the power set of X, i.e., the group of all subsets of X. Then $(X; P(X))$ is a measurable space. A set function $\mu : P(X) \to (-\infty, +\infty)$ is called a fuzzy measure or a monotone measure, if

(F1) $\mu(\emptyset) = 0$, (vanishing at the empty set)
(F2) $\mu(A) \geq 0$, for any $A \subset X$, (non-negativity)
(F3) $\mu(A) \leq \mu(B)$, if $A \subset B, A \subset X, B \subset X$, (monotonicity).

Set function μ is called an efficiency measure if it satisfies (F1) and (F2); μ is called a signed efficiency measure if it satisfies (F1) only. Any fuzzy measure is a special case of the efficiency measure; and any efficiency measure is a nonnegative set function. Fuzzy measures have a monotone constraint but efficiency measures have not, so fuzzy measures are sometimes called nonnegative monotone set functions. In multiple classifier fusion, nonnegative set functions are used to describe the importance of classifiers and the interaction among classifiers. The value of set function at a single-point-set $\mu(\{x_i\})$ presents the contribution of the single classifier x_i towards classification, and the value of set function at other sets, such as $\mu(\{x_i, x_j, x_k\})$, presents the joint contribution of classifiers towards classification. Mainly the methods to determine the nonnegative set functions have two types. One is to learn from the history data (Wang et al., 2005; Wang & Wang, 1997; Yang et al., 2008; Yeung et al., 2004) and the other is to specify by experts.

Once the set functions are available, we can use the fuzzy integral to aggregate the outputs from all classifiers. The ith column of DP matrix can be regarded as a function f_i defined on classifier set X, $f_i(x_j) = d_{j,i}$. The integral of function f_i with respect to nonnegative set function μ_i is the degree of fusion system classifying a sample to class C_i. If necessary, we can obtain the crisp class label through $C_t = \arg\max_{1 \leq i \leq c}(\int f_i d\mu_i)$.

Usually the type of fuzzy integral is chosen in advance. Choquet fuzzy integral and Sugeno fuzzy integral are often selected in fusion process. Noting that the addition and the multiplication operators are used in Choquet integrals while the maximum and the minimum operators are used in Sugeno integral, most researchers prefer now to use the Choquet integral in classifier fusion models (Wang et al., 2005). The classification process of a sample by a fused system based on fuzzy integral is illustrated in Fig. 1.

Fig. 1 shows that a sample is first submitted to all classifiers and the results from all classifiers are stored in a DP matrix. Each column of the matrix is a function defined on set X. Then the final classification result can be obtained by calculating the integral of each column of the DP matrix. The crisp class label can be finally obtained through the maximum if necessary.

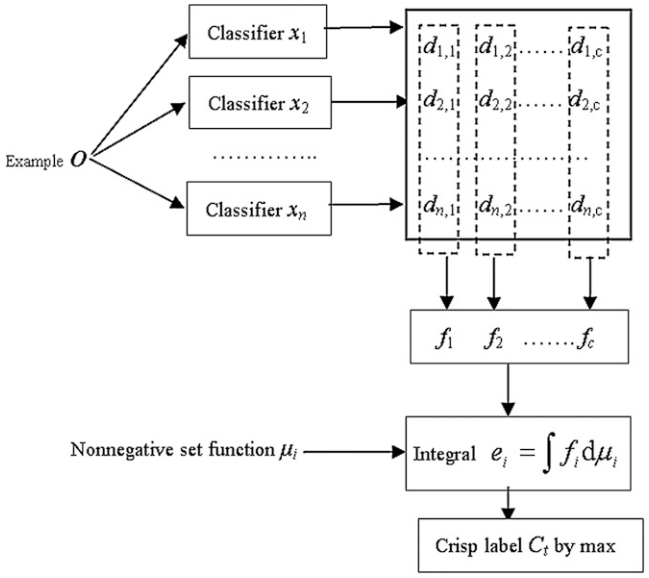

Fig. 1. The fusion system of multiple classifiers based on fuzzy integrals.

3. The upper integral and its properties

This section will introduce some mathematical concepts about the upper integral which are suitable for multiple classifier fusion.

Definition 2. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a nonempty set, $P(X)$ be the power set of X, $\mu : P(X) \to [0, +\infty)$ be a set function denoting the efficiency measure, and $f : X \to [0, +\infty)$ be a function. The upper integral of f with respect to a non-additive set-function μ is described as:

$$(U)\int f d\mu = \sup\left\{\sum_{j=1}^{2^n-1} a_j\, \mu(A_j) \mid \sum_{j=1}^{2^n-1} a_j\, \chi_{A_j} = f\right\} \tag{1}$$

where χ_{A_j} is the characteristic function of set A_j, and $a_j \geq 0$, $A_j = \bigcup_{i:j_i=1}\{x_i\}$, j is expressed in binary digits as $j_n j_{n-1} \ldots j_1$, $j = 1, 2, \ldots, 2^{n-1}$.

The value of the upper integral $(U)\int f d\mu$ is the solution of the following linear programming problem, where $a_1, a_2, \ldots, a_{2^n-1}$ are unknown parameters (Wang et al., 2008):

$$\begin{aligned}
\text{Maximum} \quad & z = \sum_{j=1}^{2^n-1} a_j \mu_j \\
\text{Subject to :} \quad & \sum_{j=1}^{2^n-1} a_j\, \chi_{A_j}(x_i) = f(x_i), \quad i = 1, 2, \ldots, n \\
& a_j \geq 0, \quad j = 1, 2, \ldots, 2^n - 1
\end{aligned}$$

where $\mu_j = \mu(A_j)$, $j = 1, 2, \ldots, 2^n - 1$. The above n constraints can be also rewritten as

$$\sum_{j:x\in A_j\subset X} a_j = f(x) \quad \forall x \in X.$$

The upper integrals have the following properties:

1. For any $c \in [0, +\infty)$, $(U)\int cf d\mu = c(U)\int f d\mu$.
2. $(U)\int f d\mu \leq (U)\int g d\mu$ if $f(x) \leq g(x)$ for every $x \in X$.
3. $(U)\int f d\mu \leq (U)\int f d\nu$ if $\mu(A) \leq \nu(A)$ for every $A \subseteq X$.
4. $(U)\int f d\mu = 0$ if and only if for every set A with $\mu(A) > 0$, there exists $x \in A$ such that $f(x) = 0$, that is, $\mu(\{x|f(x) > 0\}) = 0$.

Table 1
The values of efficiency measure μ in Example 1.

Set (combination)	Value of μ (efficiency)
$\{x_1\}$	5
$\{x_2\}$	6
$\{x_1, x_2\}$	14
$\{x_3\}$	8
$\{x_1, x_3\}$	7
$\{x_2, x_3\}$	16
$\{x_1, x_2, x_3\}$	18

Table 2
The values of function f in Example 1.

x_i	$f(x_i)$
$\{x_1\}$	10
$\{x_2\}$	15
$\{x_3\}$	7

Generally, fuzzy integrals are not linear, that is, the equality

$$(U)\int (af + bg)d\mu = a(U)\int f d\mu + b(U)\int g d\mu$$

may not be true, where a, b are two constants, f, g are integrands. Therefore, fuzzy integrals are called nonlinear integrals sometimes. For simplicity, we hide the type of integrals here. The following sample shows that the upper integral has a very intuitive and natural explanation.

Example 1 (*Wang et al., 2008*)**.** Three workers, x_1, x_2, and x_3 are engaged in producing the same kind of products. Their efficiencies (products per day) of working alone and their joint efficiencies are listed in Table 1. These efficiencies can be regarded as a nonnegative set function μ defined on the power set of $X = \{x_1, x_2, x_3\}$ with $\mu(\emptyset) = 0$ (the meaning is that there are no products if there is no worker). Here $14 = \mu(\{x_1, x_2\}) > \mu(\{x_1\}) + \mu(\{x_2\}) = 5 + 6$ means that x_1 and x_2 have a good cooperation, while $7 = \mu(\{x_1, x_3\}) < \mu(\{x_1\}) + \mu(\{x_3\}) = 5 + 8$, and even $7 = \mu(\{x_1, x_3\}) < \mu(\{x_3\}) = 8$ mean that x_1 and x_3 have a very bad relationship and they are not suitable for working together. Suppose that x_1 works for 10 days, x_2 for 15 days, and x_3 only for 7 days. Also we suppose that the manager can arrange their working in any combination, working alone or together in some way. The question now is how to arrange their working schedule such that the total products are maximized. It can be solved through the following linear programming problem.

$$\begin{aligned}
\text{Maximum} \quad & z = 5a_1 + 6a_2 + 14a_3 + 8a_4 + 7a_5 + 16a_6 + 18a_7 \\
\text{Subject to :} \quad & a_1 + a_3 + a_5 + a_7 = 10 \\
& a_2 + a_3 + a_6 + a_7 = 15 \\
& a_4 + a_5 + a_6 + a_7 = 7 \\
& a_j \geq 0, \quad j = 1, 2, \ldots, 7.
\end{aligned}$$

The optimal schedule is: x_1 and x_2 work together for 10 days, x_2 works with x_3 for 5 days, and x_3 works alone for 2 days. The relevant number of total products is 236. This value is just the upper integral $(U)\int f d\mu$ where f is listed in Table 2.

Generally, if $X = \{x_1, x_2, \ldots, x_n\}$ is the group of all workers, μ defined on $P(X)$ is the efficiency measure of the group, and f is an information function indicating the working time of each worker, then $(U)\int f d\mu$ represents the potential energy of the team (X, μ, f). The upper integral can play a role of optimization.

4. A model of classifier fusion based on upper integral

This section is to establish a new model for classifier-fusion based on the upper integral. The new model, which is totally different from the existing fuzzy integral based models, gives a sample-assignment schedule regarding how many and which samples

should be assigned to individual classifiers and their combinations, instead of the upper integral being aggregation operators.

4.1. Efficiency measure

Suppose that we are considering n classifiers, denoted by $X = \{x_1, x_2, \ldots, x_n\}$. Let $P(X)$ be the power set of X. Then each element of $P(X)$ will denote a combination of classifiers, and it is clear there are totally $2^n - 1$ combinations (excluding the empty set). For instance, $\{x_1\}$ denotes that the classifier works singly, and $\{x_1, x_3, x_4\}$ denotes the 3 classifiers work together. We first need to define an efficiency measure on $P(X)$.

Let T be the training set. Then each classifier has a training accuracy on T, and therefore, the value of the efficiency measure on a single classifier can be defined as the training accuracy, i.e., the correct rate of classification. Furthermore, suppose that we have a basic fusion operator such as average. Then, applying the fusion operator to a combination of classifiers on T, we can obtain a correct classification rate of the classifier combination on T, which is defined as the value of the efficiency measure on the classifier combination. In this way, the efficiency measure is defined as

$$\mu(A) = \begin{cases} 0 & \text{if } A = \text{ Empty set} \\ \text{Accuracy of } A \text{ on T} & \text{if } A \text{ is a nonempty subset of } X \end{cases}$$

where A denotes either a single classifier or a group of classifiers. It is worth noting that the definition of efficiency measure depends on a training-set and a basic fusion-operator for groups of classifiers.

4.2. Integrand

Since we are considering a finite space of classifiers $X = \{x_1, x_2, \ldots, x_n\}$, the integrand is a function defined on X, to be exact, a n-dimensional vector $(y_1, y_2, \ldots, y_n)$ where y_i is the proportion of samples submitted to the classifier $x_i (1 \le i \le n)$ to classify. Our goal in this subsection is to determine this integrand. Why is an assignment of samples needed? It can be seen clearly from the following example that an appropriate assignment will improve the classification accuracy in some case. Suppose there are 10 samples $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$, 3 classifiers $\{x_1, x_2, x_3\}$, the best combination $\{x_1, x_3\}$ has the highest accuracy which correctly classifies 7 samples $\{s_3, s_4, s_5, s_6, s_7, s_8, s_9\}$ and classifier, x_2, correctly classifies 6 samples $\{s_1, s_2, s_3, s_5, s_6, s_8\}$ and s_{10} cannot be correctly classified by any of combinations/classifiers. Then an appropriate assignment of samples will obtain higher accuracy than 0.8 which is the highest accuracy. Assigning $\{s_3, s_4, s_5, s_6, s_7, s_8, s_9\}$ to combination $\{x_1, x_3\}$ and $\{s_1, s_2\}$ to classifier x_2 the final accuracy is 0.9. Or assigning $\{s_4, s_6, s_7, s_8, s_9\}$ to combination $\{x_1, x_3\}$ and $\{s_1, s_2, s_3, s_5\}$ to classifier x_2 the final accuracy is 0.9 too. Under some conditions an appropriate assignment could obtain higher performance than the best combination or classifier. That is, it can improve the performance that the samples which are correctly classified by some combinations and misclassified by other combinations are assigned to appropriate combination.

Noting that the definition of upper integrals given in Section 3, we find that the value of integral expresses the highest classification efficiency for singly and jointly using classifiers $x_1, x_2, \ldots, x_n$. Specifically, the integral value denotes the highest classification efficiency and the process of computing the integral specifies a way to achieve the highest value by assigning how many samples to single classifiers and how many samples to their combinations. Here a key point we need to explicitly specify is the following. Suppose that p $(0 < p < 1)$ is the accuracy of a single classifier x_i and there exist N samples to be classified, then we will not assign all the N samples to x_i but will assign only t $(t \le pN)$ samples to x_i. It is similar to the case of a combination. Further in the next subsection, we will discuss which samples will be assigned to single classifiers and their combinations.

Assuming that the efficiency measure μ is known already, the function f can be determined by the following optimization:

$$\begin{aligned} &\text{Maximum} \quad (U)\int \{y_1, y_2, \ldots, y_n\} d\mu \\ &\text{Subject to :} \quad y_j \le \bar{\mu}_j, \quad j = 1, 2, \ldots, n \end{aligned} \tag{2}$$

where y_j denotes the proportion of samples to be assigned to classifier x_j including samples to single x_j and to combinations containing x_j, $\bar{\mu}_j = \frac{\text{samples correctly classified at least by one of combination containing } x_j}{\text{all training samples}}$ is the proportion of samples in the training set which are correctly classified by the single classifier x_j or any combination containing x_j. The inequality restriction means that samples should be assigned to a classifier or combination which can correctly classify.

The optimization problem (2) can be transferred to the following (3)

$$\begin{aligned} &\text{Maximum} \quad (U)\int \{y_1, y_2, \ldots, y_n\} d\mu = \sum_{i=1}^{2^n-1} a_i \cdot \mu_i \\ &\text{Subject to} \quad y_j = \sum_{i|b_j=1} a_i \le \bar{\mu}_j, \quad j = 1, 2, \ldots, n \\ &\qquad\qquad a_i \ge 0, \quad i = 1, 2, \ldots, 2^n - 1 \end{aligned} \tag{3}$$

where the number i has a binary expression $b_n b_{n-1} \ldots b_1$ and b_j is the jth bit; the classifier combination corresponding to a_i is $\{x_k | b_k = 1,\ k = 1, 2, \ldots, n\}$. The models (2) and (3) have such a weakness that samples for evaluating the accuracy may be counted more than once. To avoid this, we can add one more restriction:

$$\sum_{i=1}^{2^n-1} a_i = 1.$$

That is, instead of (3) we can use (4) to avoid the repeated counting of samples.

$$\begin{aligned} &\text{Maximum} \quad (U)\int \{y_1, y_2, \ldots, y_n\} d\mu = \sum_{i=1}^{2^n-1} a_i \cdot \mu_i \\ &\text{Subject to :} \quad y_j = \sum_{i|b_j=1} a_i \le \bar{\mu}_j, \quad j = 1, 2, \ldots, n \\ &\qquad\qquad \sum_{i=1}^{2^n-1} a_i = 1 \\ &\qquad\qquad a_i \ge 0, \quad i = 1, 2, \ldots, 2^n - 1. \end{aligned} \tag{4}$$

The optimization problem (4) is a linear programming problem and is easy to numerically solve. The nonzero a_i in the solution indicates the proportion of tested samples for the combination $\{x_k | b_k = 1, k = 1, 2, \ldots, n\}$ to classify. The solution of (4) results in integrand $f = \{y_1, y_2, \ldots, y_n\}$.

The classification accuracy of the upper integral based fusion system is not less than that of any individual base classifier, provided the oracles are correct. The following is a brief mathematical proof for this statement.

Proposition. *The classification accuracy of upper integral based fusion system is not less than that of any combination of classifiers, provided the oracles are correct.*

Proof. If the combination of classifiers A has the highest accuracy p, $\mu(A) = p$. Let the corresponding unknown parameter $a_A = p$, the sum of values of some other unknown parameters be $1 - p$. It is a feasible solution of optimization problem (4). If the oracles are correct, $p \times N$ tested samples are correctly classified by the combination A where N is the number of testing samples. At least the accuracy of upper integral based fusion system is $(p \times N)/N = p$. The proof is completed.

The conclusion is suitable to the case where the sum of $\bar{\mu}_j$'s is no less than 1. Note that the value of the problems (3) and (4) is not

the accuracy of the model. The value of problem (4) is $\sum_{i=1}^{2^n-1} a_i \cdot \mu_i$ which is called classification efficiency, and the accuracy of the upper integral model with correct oracles is $\sum_{i=1}^{2^n-1} a_i$.

4.3. Oracles

In Sections 4.1 and 4.2 we have discussed how to obtain the efficiency measure and the integrand for the upper-integral based classifier-fusion under the assumption that a training set and a basic fusion operator are given. In fact, the integrand gives the proportions of samples which are assigned to different combinations of classifiers. The remaining problem is which samples should be assigned to different individual classifiers and their combinations. We employ an oracle to solve this problem. Given a sample, the oracle of a combination of classifiers first predicts the possibility with which the combination can correctly classify the sample. Then the sample is allocated to the combination with maximum possibility. When the number of samples allocated to a combination attains the proportion a_i from the solution of the optimization problem (4), the allocation to this combination stops. The allocations to other combinations continue until all samples are allocated.

Practically the oracle can be obtained by training. Let T be the training set. Based on the training set T and a basic fusion operator, each combination of classifiers (including each single classifier) will have a training accuracy. Let X_C be an arbitrary combination of classifiers with accuracy p $(0 < p < 1)$. Intuitively it means that there are $(p|\mathrm{T}|)$ samples correctly classified by X_C and $((1-p)|\mathrm{T}|)$ samples incorrectly classified by X_C. Consider the $(p|\mathrm{T}|)$ samples as positive samples and the $((1-p)|\mathrm{T}|)$ samples as negative samples, we can train a new classifier which is regarded as the oracle for the combination X_C. For example, $X_C = \{x_1, x_3\}$. If the sample O is classified correctly by combination $\{x_1, x_3\}$, the target output of the oracle for the sample O will be "1". Contrarily, if the sample O is misclassified by combination $\{x_1, x_3\}$, the target output of the oracle for the sample O should be "0". Note that the correct or misclassification is based on the result of fused classification of classifiers x_1 and x_3. For unseen sample O', if the oracle corresponding combination $X_C = \{x_1, x_3\}$ output is most close to "1", we choose the classification of combination $\{x_1, x_3\}$ as system output. Summarizing the above discussions, we briefly list our scheme of upper integral based classifier fusion as follows.

Algorithm. Upper integral based classifier fusion.

Input:: T is the training set, S is the testing set, $X = \{x_1, x_2, \ldots, x_n\}$ is the group of base classifiers, F is a basic fusion operator, and L is a training algorithm for 2-class problem.

Output: Classification results of all tested samples.

Step 1: For each subset A of X, determine the efficiency measure (the accuracy of fused base classifiers in group A) based on training set T and basic operator F according to Section 4.1,

$$\mu(A) = \frac{\textit{the number of samples classified correctly by A based on F}}{\textit{the number of training samples}};$$

Step 2: Build and solve the optimization problem (4) given in Section 4.2 to determine the integrand. According to the solution of problem (4), combinations (including single base classifier) corresponding to nonzero a_i are chosen to classify unseen samples, and a_i is the proportion of samples in testing set S that is assigned to the corresponding combination;

Step 3: Train oracles for all chosen combinations by using algorithm L according to the 2nd in Section 4.3;

Step 4: According to the oracles trained in Step 3, a sample in S is assigned to the combination which has not reached its proportion and has the highest possibility to correctly classify the sample;

Step 5: Let the combination from Step 4 classify the assigned sample based on F;

Step 6: Calculate the final classification results.

Step 7: Repeat Steps 4–6 until all samples in S are classified.

It is worth noting that, the base classifiers are assumed to be known in advance.

Table 3
Specification of classification data sets and the number of hidden neurons for SLFNs in the experiment.

Data sets	# Attributes	# Classes	# Samples
Iris	5	3	150
Breast cancer	10	2	683
Tic-Tac-Toe	9	2	958
Ionosphere	34	2	351
Pima	8	2	768
Heart	10	2	270
Wine	13	3	178
Sonar	60	2	208
Letter	17	26	20 000
Waveform + noise	41	3	5 000

5. Experiment results

5.1. Comparison with Boosting/Bagging

In order to know well the upper integral based model of fusion, an empirical study is performed in this section. Ten benchmark data sets are respectively selected from UCI machine learning repository (UCI, 0000). They are sized from 150 to 20 000, and the detailed information is summarized in Table 3.

All the simulations are carried out in MATLAB 2007 environment running on an Intel T2400, 1.83 GHz CPU. The upper integral fusion model is compared with Breiman (1996) and Freund and Schapire (1997). In our experiment, 10-fold cross validation is repeated 20 times on each data set. Both Bagging and Boosting contain 100 base classifiers. When the upper integral model is compared with Bagging, the upper integral model uses the 100 base classifiers trained by Bagging and the basic fusion operator is majority vote. Similarly, in comparing with Boosting the upper integral model uses the 100 base classifiers trained by Boosting and the basic fusion operator is weighted majority vote where the weights are determined during the training of Boosting. Here the upper integral only considers the single base classifiers and combinations which consist of two or three base classifiers (not all possible combinations). The parameters in efficiency measure increase exponentially with the number of base classifiers. When the number of base classifiers is small, we can consider all the combinations. It is difficult to find the optimal solution of problem (4) when it is large. It is needed to balance the accurate solution and the feasibility.

The 4 types of base classifiers, i.e., ELM, conventional back-propagation single-layer neural networks (BP), ELM with Gaussian kernel (ELM-kernel) (Huang, Zhou, Ding, & Zhang, 2012), and least square support vector machine (LS-SVM, available at http://www.esat.kuleuven.ac.be/sista/lssvmlab/) are respectively implemented in the verification experiment. For the ELM and BP, the transfer function is fixed as hyperbolic tangent sigmoid. The number of hidden neurons used in each data set is determined by a ten-fold cross validation on each data set. The number which achieves the best average cross-validation accuracy will be selected for ELM and BP respectively. Through observing the performance with different numbers, the appropriate step, such 5, 10, 50, can be adopted to search the best number of hidden neurons. In general, the performance will quickly increase with more hidden neurons, so the step could be larger. When the growth of accuracy slowed, a smaller step is adopted. When a turning-point appears, the number of hidden neurons is chosen as the best. In order to achieve good generalization performance, the cost parameter C and kernel parameter γ of ELM-kernel and LS-SVM need to be chosen appropriately. For each data set, we have used 50 differ-

Table 4
Comparison of three fusion schemes for ELM and BP (the correct rate).

Data	ELM				BP			
	Upper integral based Boosting	Boosting	Upper integral based Bagging	Bagging	Upper integral based Boosting	Boosting	Upper integral based Bagging	Bagging
Iris	**0.9839**	0.9728	0.9831	0.9696	0.9481	0.9377	0.9565	0.9367
Breast	**0.9591**	0.9472	0.9529	0.9394	0.9312	0.9172	0.9310	0.9138
Tic-Tac-Toe	0.9247	0.9102	**0.9321**	0.9001	0.9216	0.9056	0.9251	0.8914
Ionosphere	0.8566	0.8301	**0.8596**	0.8313	0.8369	0.8194	0.8330	0.8217
Pima	**0.7912**	0.7686	0.7891	0.7532	0.7493	0.7287	0.7384	0.7181
Heart	0.8438	0.8273	**0.8451**	0.8191	0.8215	0.7967	0.7962	0.7771
Wine	0.9418	0.9281	**0.9469**	0.9327	0.9287	0.9016	0.9188	0.8962
Sonar	0.8567	0.8289	**0.8626**	0.8310	0.8207	0.8003	0.8167	0.7983
Letter	**0.9519**	0.9307	0.9462	0.9265	0.9172	0.9002	0.9207	0.8963
Waveform + noise	**0.8528**	0.8319	0.8487	0.8261	0.8231	0.8173	0.8279	0.8031

Table 5
Comparison of three fusion schemes for ELM-kernel and LS-SVM (the correct rate).

Data	ELM-kernel				LS-SVM			
	Upper integral based Boosting	Boosting	Upper integral based Bagging	Bagging	Upper integral based Boosting	Boosting	Upper integral based Bagging	Bagging
Iris	**0.9867**	0.9818	0.9868	0.9791	0.9840	0.9848	0.9721	0.9672
Breast	0.9731	0.9519	0.9687	0.9421	**0.9811**	0.9569	0.9756	0.9512
Tic-Tac-Toe	**0.9473**	0.9123	0.9369	0.9316	0.9252	0.9105	0.9312	0.9123
Ionosphere	0.8901	0.8672	**0.8965**	0.8758	0.8749	0.8470	0.8629	0.8427
Pima	**0.8015**	0.7826	0.7971	0.7763	0.7992	0.7768	0.7868	0.7529
Heart	0.8672	0.8429	**0.8722**	0.8511	0.8709	0.8321	0.8531	0.8247
Wine	**0.9887**	0.9856	0.9818	0.9796	0.9869	0.9835	0.9872	0.9813
Sonar	0.8824	0.8526	0.8768	0.8449	0.8794	0.8376	**0.8830**	0.8427
Letter	0.9839	0.9775	**0.9881**	0.9768	0.9773	0.9556	0.9801	0.9610
Waveform + noise	0.8774	0.8496	**0.8887**	0.8687	0.8808	0.8521	0.8783	0.8592

ent values of C and 50 different values of γ, resulting in a total of 2500 pairs of (C, γ). The 50 different values of C and γ are $\{2^{-24}, 2^{-23}, \ldots, 2^{24}, 2^{25}\}$. We conduct a ten-fold cross validation on each data set and select the pair of (C, γ) for ELM-kernel and LS-SVM respectively, which achieves the best average cross-validation accuracy. The results are shown in Tables 4 and 5.

As seen from Table 4, the performance of three fusion systems, the upper integral, Bagging and Boosting, with ELMs is higher than that of fusion system with BPs on 10 data sets. This is in conformity with the conclusions in Huang et al. (2006). Table 5 shows that the accuracies with ELM-kernel are higher or similar to those with LS-SVM. It tallies with the result in Huang et al. (2012). Also it shows that the performance of fusion system is dependent on the performance of base classifier. The performance of the upper integral model is higher or similar to that of Bagging/Boosting on 10 data sets. It shows that the upper integral model can obtain higher or similar performance to that of Bagging/Boosting. It demonstrates that the upper integral model could capture the interaction between base classifiers and make good use of the interaction through assigning tested samples to different individual classifiers and their combinations. That is, the upper integral could be used to improve existing fusion model. The base classifiers in Boosting have stronger interaction than those in Bagging. In application, the base classifiers could be designed through other way and the basic fusion operator can be others.

5.2. Comparison with existing fuzzy integral models

In this subsection we experimentally compare our approach with existing fuzzy integral models. The basic fusion operator in our approach is the average and the upper integral is used to improve the classifier fusion system by assigning tested samples to different classifier groups. Choquet integral, which is a type of most frequently used fuzzy integrals due to its simplicity and availability (Wang et al., 2008), is here selected as the fusion operator in comparison with our approach. Three methods are used to determine the fuzzy measures for Choquet integral model. The first one (written as λ-measure 1) is the λ-measure determined according to following Eq. (5) (Verikas et al., 1999):

$$g^i = \frac{p_i}{\sum_{j=1}^{n} p_j} \tag{5}$$

where p_i is the accuracy of the ith classifier. The λ-measure is used for all classes.

The second (written as λ-measure 2) is λ-measures determined according to following Eq. (6) (Verikas et al., 1999):

$$g^{ij} = \frac{p_{ij}}{\sum_{t=1}^{n} p_{tj}} \tag{6}$$

where p_{ij} is the accuracy of the ith classifier classifying samples from j class. The λ-measure $g^j = \{g^{1j}, g^{2j}, \ldots, g^{nj}\}$ is used for determining the possibility of samples belonging to jth class.

The third method to determine fuzzy measure is the genetic algorithm (Yang et al., 2008; Zhai et al., 2013). The population size is 100. The genetic algorithm is used to determine λ-measure (written as GAλ) and regular fuzzy measure ($2^n - 1$ unknown parameters, written as GAregular).

Ten-fold cross validation is repeated 20 times on each data set. Ten ELMs are trained as the base classifier for both Choquet integral and our approach. The results are listed in Table 6.

From Table 6 one can view that (1) the training for λ-measures 1 and 2 is extremely fast; (2) the training time of upper integral model is much shorter than that of genetic algorithms; (3) the performance of λ-measure 1 is the lowest; (4) the performance of λ-measure 2 is better than that of λ-measure 1 but worse than that of λ-measure determined by genetic algorithm; (5) the performance of upper integral is the highest on 7 out of 10 data sets; and (6) the performance of regular fuzzy measure determined

Table 6
Comparison of upper integral model with existing fuzzy integral models (the correct rate and the training time).

Data	Upper integral		λ-measure 1		λ-measure 2		GAλ		GAregular	
	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)
Iris	0.956	4	0.928	0.1	0.934	0.1	0.949	57	**0.957**	21
Breast	**0.894**	9	0.841	0.1	0.850	0.1	0.860	167	0.874	132
Tic-Tac-Toe	**0.937**	9	0.887	0.1	0.905	0.1	0.911	419	0.921	423
Ionosphere	**0.808**	5	0.748	0.1	0.767	0.1	0.778	138	0.788	111
Pima	**0.708**	6	0.655	0.1	0.671	0.1	0.682	201	0.692	171
Heart	**0.712**	8	0.673	0.1	0.672	0.1	0.686	109	0.697	89
Wine	0.754	4	0.725	0.1	0.737	0.1	0.746	77	**0.755**	45
Sonar	0.793	5	0.771	0.1	0.772	0.1	0.784	61	**0.797**	30
Letter	**0.875**	16	0.825	0.1	0.848	0.1	0.854	892	0.867	555
Waveform + noise	**0.781**	11	0.745	0.1	0.761	0.1	0.770	743	0.773	411

by genetic algorithm is highest on 3 out of 10 data sets (but their time complexity of training is much higher than our approach). Moreover it is worth noting that the time complexity for training GA-regular fuzzy measures is exponentially increasing with the number of base classifiers. Considering both the accuracy and the training complexity, we experimentally validate that our approach is superior to the fusion model based on Choquet integral.

6. Conclusions and discussions

This paper proposes a multiple classifier fusion method based on the upper integral to most effectively use the individual ELMs and their combinations. The difficulty of determining the fuzzy measures is avoided by regarding the accuracies of classifier combinations as an efficiency measure defined on the power set of classifier set. The upper integral is used to determine the proportions of samples to be assigned to classifier combinations instead of aggregation operator. Through solving an optimization problem with respect to the upper integral, the proportions can be obtained. According to these proportions and some trained oracles, the assignment is conducted. Theoretically, the definition of upper integrals indicates that the accuracy of upper integral based fusion system is not lower than that of any combination of classifiers. The experiment results show that the upper integral based fusion approach can improve the performance with ELMs as base classifiers. In comparison with Bagging/Booting and fuzzy integral fusion models, our proposed upper integral model can obtain a better performance in most cases. It demonstrates that the upper integral model could capture the interaction between ELMs and make good use of the interaction through assigning tested samples to different individual ELMs and their combinations.

Our proposed scheme is dependent on the testing data. Each time, the scheme simultaneously handles the classification of a batch of samples rather than a sample. To some extent it may limit the applicability. But for such real applications where the samples are coming batch by batch (rather than one by one), the scheme will have its significant advantages of higher accuracy (as the paper shows). One reason is that the batch not only includes individual samples but also includes their relationships. In real applications we often see the classification problems in which samples are coming in batch. For example, in remote image processing, often we obtain a batch of images for classification from different sensors simultaneously.

Acknowledgements

This research is supported by the Natural Science Foundation of China (61170040 and 71371063) and by Hebei Natural Science Foundation (F2013201110).

References

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.
Freund, Y., & Schapire, R. E. (1997). A decision–theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.
Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001.
Huang, G.-B., Chen, L., & Siew, C.-K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, *17*(4), 879–892.
Huang, G.-B., Zhou, H., Ding, X., & Zhang, Rui (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *42*(2), 513–529.
Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. *International Joint Conference on Neural Networks*, *2*, 985–990.
Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*, 489–501.
Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37.
Keller, J. M., Gader, P., Tahani, H., Chiang, J.-H., & Mohamed, M. (1994). Advances in fuzzy integration for pattern recognition. *Fuzzy Sets and Systems*, *65*, 273–283.
Kuncheva, L. I. (2003). *Combining pattern classifiers: methods and algorithms*. Hoboken, New Jersey: A Wiley-Interscience Publication.
Lan, Y., Soh, Y. C., & Huang, G.-B. (2009). Ensemble of online sequential extreme learning machine. *Neurocomputing*, *72*(13), 3391–3395.
Liu, Q., He, Q., & Shi, Z. (2008). Extreme support vector machine classifier, In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 222–235).
Liu, N., & Wang, H. (2010). Ensemble based extreme learning machine. *IEEE Signal Processing Letters*, *17*(8), 754–757.
Schmitt, E., Bombardier, V., & Wendling, L. (2008). Improving fuzzy rule classifier by extracting suitable features from capacities with respect to the Choquet integral. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *38*(5), 1195–1206.
Shafer, G. (1976). *A mathematical theory of evidence*. Princeton, New Jersey, USA: Princeton University Press.
UCI Repository of Machine Learning Databases and Domain Theories. Available: [Online] http://www.ics.uci.edu/mlearn/MLRepository.html.
Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., & Gelzinis, A. (1999). Soft combination of neural classifiers: a comparative study. *Pattern Recognition Letters*, *20*, 429–444.
Wang, X. Z., Chen, A. X., & Feng, H. M. (2011). Upper integral network with extreme learning mechanism. *Neurocomputing*, *74*(16), 2520–2525.
Wang, Z., Leung, K.-S., & Klir, G. J. (2005). Applying fuzzy measures and nonlinear integrals in data mining. *Fuzzy Sets and Systems*, *156*(3), 371–380.
Wang, Z., Li, W., & Leung, K.-S. (2008). Lower integrals and upper integrals with respect to nonadditive set functions. *Fuzzy Sets and Systems*, *159*, 646–660.
Wang, J., & Wang, Z. (1997). Using neural networks to determine Sugeno measures by statistics. *Neural Networks*, *10*(1), 183–197.
Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, *18*, 183–190.
Yang, R., Wang, Z., Heng, P.-A., & Leung, K.-S. (2008). Fuzzified Choquet integral with a fuzzy-valued integrand and its application on temperature prediction. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *38*(2), 367–380.
Yeung, D. S., Wang, Xi-Zhao, & Tsang, E. C. C. (2004). Handling interaction in fuzzy production rule reasoning. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *34*(5), 1979–1987.
Zhai, J., Xu, H., & Li, Y. (2013). Fusion of extreme learning machine with fuzzy integral. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *21*(supp02), 23–34.
Zhai, J., Xu, H., & Wang, X. (2012). Dynamic ensemble extreme learning machine based on sample entropy. *Soft Computing*, *16*(9), 1493–1502.