A New Approach to Classifier Fusion Based on Upper Integral

Xi-Zhao Wang, *Fellow, IEEE,* Ran Wang, *Student Member, IEEE,* Hui-Min Feng, and Hua-Chao Wang

***Abstract*—Fusing a number of classifiers can generally improve the performance of individual classifiers, and the fuzzy integral, which can clearly express the interaction among the individual classifiers, has been acknowledged as an effective tool of fusion. In order to make the best use of the individual classifiers and their combinations, we propose in this paper a new scheme of classifier fusion based on upper integrals, which differs from all the existing models. Instead of being a fusion operator, the upper integral is used to reasonably arrange the finite resources, and thus to maximize the classification efficiency. By solving an optimization problem of upper integrals, we obtain a scheme for assigning proportions of examples to different individual classifiers and their combinations. According to these proportions, new examples could be classified by different individual classifiers and their combinations, and the combination of classifiers that specific examples should be submitted to depends on their performance. The definition of upper integral guarantees such a conclusion that the classification efficiency of the fused classifier is not less than that of any individual classifier theoretically. Furthermore, numerical simulations demonstrate that most existing fusion methodologies, such as bagging and boosting, can be improved by our upper integral model.**

***Index Terms*—Efficiency measure, fuzzy integral, interaction, multiple classifier fusion, nonadditive set function, upper integral.**

I. Introduction

FUZZY INTEGRALS [1], [2] have been widely applied in classification problems, such as computer vision [3], intrusion detection [4], and biotechnology [5]. Since fuzzy measures used in fuzzy integrals can model and represent the interactions among classifiers or attributes, fuzzy integrals usually have better performance than other classification schemes when attributes of the problem are strongly of interaction. There are two points of view for applying fuzzy integrals in classification problems: one is the fuzzy pattern matching [6], [7] and the other is the multiple classifier fusion [9]–[11]. The former considers the fuzzy integral as a classifier, while the later considers the fuzzy integral as a fusion operator. The fuzzy integral classifier is based on a fuzzy pattern-matching process including two steps: 1) computing the matching degree between a testing example and the fuzzy prototype in which the attributes are represented as fuzzy subsets, and 2) merging all matching degrees concerning a class into a single value by fuzzy integrals. The fusion is usually conducted through the following three steps: 1) the individual classifier outputs the membership degrees of an example belonging to different classes; 2) the fuzzy integral is used to combine all these membership degrees; and 3) the class with the highest fused membership degree is chosen as the final classification result. Fuzzy integral based fusion of multiple classifiers has been one hot topic in machine learning during the recent decades.

Many types of classifiers, e.g., decision trees [12]–[14], neural networks [14], support vector machines [14], [15], have been proposed in the recent decades, but it is difficult to say which model is the best for a specific task [10]. In many practical tasks, it is difficult to design a single classifier with satisfying performance. Considering the neural network, although Hornik [16] theoretically proved that multilayer feed-forward networks with one hidden layer using arbitrary squashing functions (e.g., sigmoid functions) are capable of approximating any continuous function given a sufficient number of hidden units; practically, it is almost impossible to design a neural network with the optimal units. Hasen and Salamon [17] showed that the generalization ability could be significantly improved through fusing a number of neural networks, i.e., training many neural networks and then combining their predictions. Since the fusion technology is easy and behaves remarkably well, it has been successfully applied to many areas [18]–[21], and the types of classifiers can not only be neural networks but also be decision trees, k-nearest neighbors, etc.

In multiple classifier fusion, a number of base classifiers are first designed for a given classification task. Then, a fusion operator, such as minimum, maximum, median, average, weighted average, ordered weighted average, or fuzzy integral, is selected to aggregate the outputs from all base classifiers [17]–[21]. The aggregated results are the final classification. Weighted average and ordered weighted average operators are good choices to deal with the different importance of base

Manuscript received December 18, 2011; revised October 26, 2012, January 28, 2013, and April 21, 2013; accepted May 6, 2013. Date of publication June 13, 2013; date of current version April 11, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61170040 and Grant 60903089, the Natural Science Foundation of Hebei Province under Grant F2011201063, Grant F2012201023, Grant F2013201110, Grant F2013201060, and Grant F2013201220, and the Key Scientific Research Foundation of Education Department of Hebei Province under Grant ZD2010139. This paper was recommended by Associate Editor S.-F. Su.

X.-Z. Wang and H.-M. Feng are with the Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding 071002, Hebei, China (e-mail: xizhaowang@ieee.org; hmfeng@hbu.edu.cn).

R. Wang is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: ranwang3-c@my.cityu.edu.hk).

H.-C. Wang is with the National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China (e-mail: huachao1001@163.com).

Digital Object Identifier 10.1109/TCYB.2013.2263382

classifier, but the two methods are based on an assumption that there is no interaction between the base classifiers. However, this assumption may not be true in many real problems. If the interaction is considered, the Dempster–Shafers method [20], [22]–[25] or fuzzy integrals [5]–[8], [18] may be a better choice. It was shown in [22] that the fuzzy integral is more practical than the Dempster–Shafers method. The fuzzy integral as a fusion tool, in which the nonadditive measure can clearly express the interaction among classifiers and the importance of each individual classifier, has particular advantages. One difficulty for applying fuzzy integrals is how to estimate the fuzzy measures. There are some methods to determine fuzzy measures such as linear programming, quadratic programming [26], [27], genetic algorithm [28], [29], neural network [30], and pseudogradient [31].

This paper proposes a new approach to multiple classifier fusion based on the upper integral. A new type of fuzzy integral was proposed in [42]. This new integral was then extended to a pair of integrals, i.e., the lower and the upper integrals in [32]. In addition, when the universal set is finite, the upper integral is called Wang integral and its calculation method was shown in [42].

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

The rest of this paper is arranged as follows. In Section II, the related work is introduced. The existing multiple classifier fusion schemes are reviewed in Section III. Section IV is devoted to the efficiency measures, fuzzy integrals, and upper integrals. Our new fusion scheme based on the upper integral is proposed in Section V. Section VI presents a number of numerical experiments to verify advantages of the new approach, and finally Section VII concludes this paper.

II. Related Work

There are several reasons for combining multiple classifiers to solve a given classification problem [33]. For example, different classifiers trained on the same data may not only differ in their global performances, but they also may show strong local differences. Each classifier may have its own region in the feature space where it performs best. Most combination schemes in the literature belong to parallel architecture in which all the base classifiers are invoked independently, and a fusion operator then fuses their outputs [33].

The outputs of the base classifiers are usually imprecise or uncertain. To handle/fuse the uncertain or imprecise information, we can find many useful approaches, theories, and operators, such as Bayesian method [20], [33], Dempster–Shafer evidence theory [20], [22]–[25] and fuzzy integrals [7]–[9], [11], [19]–[21], [26], [27], [29], [31], [34].

1) The Bayesian method describes the uncertainty and completes the inference based on the prior and posteriori probability. Theoretically, the Bayes decision rule gives the optimal classification correction rate, in the sense that, for a given (A) prior probability, (B) loss function, and (C) class-conditional density, no other decision rule will have a lower risk. But, practically, Bayes methods often have the poor performance. One reason is that the basic assumption in Naive Bayes, i.e., the independence of classifiers, is often not satisfied. Another reason is that it is difficult to precisely estimate the class-conditional densities or prior probabilities because of absence of some necessary information in many classification tasks [24].
2) The Dempster–Shafer approach gives a representation of imprecise and uncertain results from classifiers through two sets of functions: plausibility and belief [22], [24]. With belief theory, each classifier first generates a belief function over the power set of classes and the outputs are combined by using Dempster's rule.
3) The fuzzy integral, as an extension of average operator, can combine the outputs of base classifiers and their combinations. The values of fuzzy integrals provide a measure of certainty for classification, and this measure significantly differs from the well-known posteriori probabilities. A key problem for applying fuzzy integrals is how to determine suitable fuzzy measures. From [3], [4], [7], [11], [26]–[31], and [39]–[41], one can find several existing methods of determining fuzzy measures. Fuzzy integrals are more computationally efficient than a strict Dempster–Shafer approach [25].

Moreover, bagging [35] and boosting [36], which are considered an ensemble meta-algorithm for improving the stability and accuracy of learning algorithms used in classifications

and regressions, also have a mechanism of fusing uncertainty. Different from the Bayesian and DS theory, bagging and boosting operate by taking a base learning algorithm and invoking it many times with different training sets. Bagging and Adaboosting are also techniques widely used to build diverse classifiers. The diversity is an important factor for a successful fusion system [20], [37], [38].

We now focus on a new type of fuzzy integrals, called the upper integral. The upper integral can be interpreted as the maximum efficiency under certain restraint conditions. This paper makes an attempt to shift this concept of maximum efficiency to the classification problem based on the upper integrals.

III. 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_j 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 example 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 an example can be organized in the decision profile matrix $DP = [d_{j,i}]_{n\times c}$ [20].

Each column of the DP matrix can be regarded as a function defined on the 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 toward examples from C_i class. Set functions have some special cases.

Definition 1 [43]: Let X be a nonempty and finite set and $P(X)$ be the power set 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:

(FM1) $\mu(\emptyset) = 0$, (vanishing at the empty set);

(FM2) $\mu(A) \geq 0$, for any $A \subset X$, (non-negativity);

(FM3) $\mu(A) \leq \mu(B)$, if $A \subset B$, $A \subset X$, $B \subset X$, (monotonicity).

Set function μ is called an efficiency measure if it satisfies (FM1) and (FM2); μ is called a signed efficiency measure if it satisfies (FM1) 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 toward classification, and the value at other sets, such as $\mu(\{x_i, x_j, x_k\})$, presents the joint contribution of the three classifiers toward classification. Mainly the ways to determine the nonnegative set functions have two types. One is to learn from the history data [11], [26]–[31] and the other is to specify by experts.

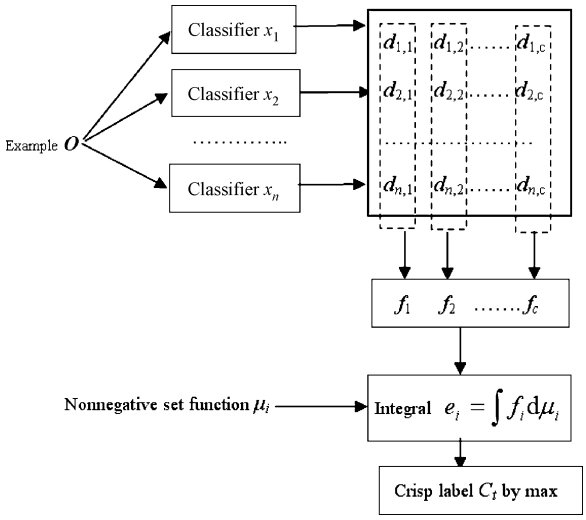


Fig. 1. Fusion system of multiple classifiers based on fuzzy integrals.

Once the set functions are available, we can use the fuzzy integral to aggregate the outputs from all classifiers. The integral of function f_i (the ith column of DP matrix) with respect to nonnegative set function μ_i is the degree of fusion system classifying an example 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. Noting that the addition and the multiplication operators are used in Choquet integral while the maximum and the minimum operators are used in Sugeno integral, most researchers prefer to choose Choquet integral [32]. The classification process of an example by a fused system based on fuzzy integrals is illustrated in Fig. 1.

Fig. 1 shows that an example 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.

IV. Upper Integral and Its Properties

This section will introduce some mathematical concepts about fuzzy integrals that are suitable for multiple classifier fusion.

Consider a nonempty set $X = \{x_1, x_2, \ldots, x_n\}$. $P(X)$ is the power set of X, $\mu : P(X) \to [0, +\infty)$ is an efficiency measure, and $f : X \to [0, +\infty)$ is a function. First, $\{f(x_1), f(x_2), \ldots, f(x_n)\}$ are rearranged into a nondecreasing order, that is

$$f(x_1^*) \leq f(x_2^*) \leq \ldots \leq f(x_n^*)$$

where $\{x_1^*, x_2^*, \ldots, x_n^*\}$ is a permutation of $\{x_1, x_2, \ldots, x_n\}$. Then, the Choquet integral and the Sugeno integral of function f with respect to measure μ are respectively evaluated as

$$(C)\int f d\mu = \sum_{i=1}^{n}\left[f(x_i^*) - f(x_{i-1}^*)\right]\cdot \mu(\{x_i^*, x_{i+1}^*, \ldots, x_n^*\}) \tag{1}$$

where $f(x_0^*) = 0$

$$(S)\int f d\mu = \max_{1\le i\le n}\left[f(x_i^*) \wedge \mu(\{x_i^*, x_{i+1}^*, \ldots, x_n^*\})\right]. \tag{2}$$

Choquet integral and Sugeno integral are two aggregation operators often used in multiple classifier fusion. In application, set X consists of all classifiers. The supports of one example belonging one class from all classifiers, the column of a DP matrix, constitute the integrand of fuzzy integral. In additional, another important fuzzy integral, called the upper integral, has been proposed in [32] and [42]. The upper integral of f with respect to a non-additive set-function μ is described as (5)

$$(U)\int f d\mu = sup\{\sum_{j=1}^{2^n-1} a_j\,\mu(A_j) \mid \sum_{j=1}^{2^n-1} a_j\,\chi_{A_j} = f\} \tag{3}$$

where χ_{A_j} is the characteristic function of set A_j, and $a_j \ge 0$, $A_j = \bigcup_{i:j_i=1}\{x_i\}$, 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 [32], [42]:

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

where $\mu_j = \mu(A_j)$, $j = 1, 2, \ldots, 2^n - 1$, $a_1, a_2, \ldots, a_{2^n-1}$ are unknown parameters. The above n constraints can be also rewritten as

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

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) $(U)\int f d\mu \le (U)\int g d\mu$ if $f(x) \le g(x)$ for every $x \in X$.
3) $(U)\int f d\mu \le (U)\int f d\nu$ if $\mu(A) \le \nu(A)$ 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, $\mu(\{x|f(x) > 0\}) = 0$.
5) $(C)\int f d\mu \le (U)\int f d\mu$.

Since the integral value represents an efficiency of classification, which is considered the classification accuracy when the oracle is 100% correct, (5) indicates that the upper integral model is more efficient than the Choquet integral model regarding classification problems.

V. Model of Classifier Fusion Based on Upper Integral

The aim of this section is to establish a new model for classifier-fusion based on the upper integral. The new model, which is completely different from the exiting fuzzy integral based models, gives an example-assignment schedule regarding how many and which examples should be assigned to individual classifiers and their combinations, instead of being aggregation operators.

A. 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, i.e., the group of all subsets of X. Then, each element of $P(X)$ will denote a combination of classifiers, and it is clear there are $2^n - 1$ combinations in total (excluding the empty set). For instance, $\{x_1\}$ denotes that the classifier works alone, and $\{x_1, x_3, x_4\}$ denotes the three 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 majority voting or 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 efficient 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.

B. 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, an n-dimensional vector $(y_1, y_2, \ldots, y_n)$ where y_i is the proportion of examples submitted to the classifier $x_i (1 \le i \le n)$ to classify. Our goal in this subsection is to determine this integrand.

Noting the definition of upper integrals given in Section IV, 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 examples to single classifiers and how many examples 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 examples to be classified, then we will not assign all the N examples to x_i but will assign only t $(t \le pN)$ examples to x_i. It is similar to the case of a mixture of classifiers. Further in the next subsection, we will discuss which examples will be assigned to single classifiers and their combinations.

Assuming that the integrand to be determined is expressed as $f = \{y_1, y_2, \ldots, y_n\}$ and the efficiency measure μ is known already. Then, the function f can be determined by the following optimization:

$$\begin{array}{ll} \text{Maximum} & (U)\int\{y_1, y_2, \ldots, y_n\}d\mu \\ \text{Subject to} & y_j \le \mu_j,\ j = 1, 2, \ldots, n \end{array} \tag{4}$$

where y_j denotes the proportion of examples assigned to classifier x_j, μ_j is the value of the efficiency measure on the single classifier x_j. The inequality restriction means that the proportion of examples assigned to each individual classifier should not exceed the correct rate (accuracy) of the classifier.

The optimization problem (4) can be transferred to

$$\begin{array}{ll} \text{Maximum} & (U)\displaystyle\int\{y_1, y_2, \ldots, y_n\}d\mu = \sum_{i=1}^{2^n-1} a_i \cdot \mu_i \\ \text{Subject to} & y_j = \displaystyle\sum_{i|b_j=1} a_i \le \mu(\{x_j\}),\ j = 1, 2, \ldots, n \\ & a_i \ge 0, i = 1, 2, \ldots, 2^n - 1 \end{array} \tag{5}$$

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 $\{x_k | b_k = 1,\ k = 1, 2, \ldots, n\}$. Models (4) and (5) have such a weakness that examples 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 (5) we can use (6) to avoid the repeat counting examples

$$\begin{array}{ll} \text{Maximum} & (U)\displaystyle\int\{y_1, y_2, \ldots, y_n\}d\mu = \sum_{i=1}^{2^n-1} a_i \cdot \mu_i \\ \text{Subject to} & y_j = \displaystyle\sum_{i|b_j=1} a_i \le \mu(\{x_j\}),\ j = 1, 2, \ldots, n \\ & \displaystyle\sum_{i=1}^{2^n-1} a_i = 1 \\ & a_i \ge 0, i = 1, 2, \ldots, 2^n - 1. \end{array} \tag{6}$$

The optimization problem (6) is a linear programming problem and is easy to numerically solve. The nonzero a_i in the solution indicates the proportion of testing examples for the combination $\{x_k | b_k = 1,\ k = 1, 2, \ldots, n\}$ to classify. The solution of (6) results in integrand $f = \{y_1, y_2, \ldots, y_n\}$.

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

Proposition: The integral value in optimization problem (6) is not less than the classification accuracy of any individual base classifier if oracles are correct. That is

$$(U)\int\{y_1, y_2, \ldots, y_n\}d\mu \ge \mu(\{x_j\}),\ j = 1, 2, \ldots, n.$$

Proof: If the base classifier x_{i^*} has the highest accuracy p^*, $\mu(\{x_{i^*}\}) = p^*$. Let the corresponding unknown parameter $a_i^* = p^*$, one of the other unknown parameters, be $1 - p^*$ where the accuracy of the corresponding classifier is not less than $1 - p^*$, and let all the rest unknown parameters be zero. It is a feasible solution of the optimization problem (6). If the oracles are correct, $p^* \times N$ testing examples are correctly classified by the base classifier x_{i^*} where N is the number of testing examples. At least the accuracy of the upper integral based fusion system is $(p^* \times N)/N = p^*$, $(U)\int\{y_1, y_2, \ldots, y_n\}d\mu \ge \mu(\{x_j\}),\ j = 1, 2, \ldots, n$. The proof is completed.

C. Oracles

In Sections V-A and V-B, 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 is given. In fact, the integrand gives the proportions of examples that are assigned to different combination of classifiers. Obviously, according to the property of upper integral, the value of integral is not less than the classification accuracy of any individual classifier. This indicates that, following the assignment proportions determined by the integrand, the classification efficiency of the upper-integral-based fusion can achieve the highest value. The remaining problem is which examples should be assigned to different individual classifiers and their combinations. We employ an oracle to solve this problem. Given an example, the oracle of a combination of classifiers first predicts the possibility with which the combination can correctly classify the example. Then, the example is allocated to the combination with maximum possibility. When the number of examples allocated to a combination attains the proportion a_i from the solution of the optimization problem (6), the allocation to this combination stops. The allocations to other combinations continue until all examples 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 $(p|\mathrm{T}|)$ examples correctly classified by X_C and $((1-p)|\mathrm{T}|)$ examples incorrectly classified by X_C. Consider the $(p|\mathrm{T}|)$ examples as positive examples and the $((1-p)|\mathrm{T}|)$ examples as negative examples, 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 example O is classified correctly by combination $\{x_1, x_3\}$, the target output of the oracle for the example O should be 1. Contrarily, if the example O is misclassified by combination $\{x_1, x_3\}$, the target output of the oracle for the example O should be 0. Note that the correct or misclassification is based on the fused result from classifiers x_1 and x_3. For unseen example O', if the output of the oracle corresponding combination $X_C = \{x_1, x_3\}$ is most close to 1, we choose the classification of combination $\{x_1, x_3\}$ as system output. Summarizing the above discussions, we list our scheme of upper integral based classifier fusion as follows.

TABLE I

CLASSIFICATION OF THREE CLASSIFIERS AND THEIR COMBINATIONS ON A DATASET WITH 20 EXAMPLES: + DENOTES CORRECT CLASSIFICATION AND - DENOTES INCORRECT CLASSIFICATION

Example No.	Class	Individual classifiers and their combinations						
		$\{x_1\}$	$\{x_2\}$	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_1, x_3\}$	$\{x_2, x_3\}$	$\{x_1, x_2, x_3\}$
1	1	-	+	+	-	-	-	-
2	0	+	+	+	+	+	+	+
3	1	+	+	+	+	+	+	+
4	1	+	+	+	+	+	+	+
5	1	-	-	-	+	+	+	-
6	0	+	+	+	-	+	+	+
7	1	+	+	+	-	+	-	+
8	0	+	+	+	+	+	+	-
9	1	+	+	+	+	+	+	+
10	0	+	+	+	+	+	+	+
11	0	+	+	+	+	+	+	+
12	0	+	+	+	+	+	+	+
13	1	+	+	+	+	+	+	+
14	1	+	+	+	+	+	+	+
15	0	-	-	-	-	+	-	-
16	0	-	-	-	-	+	-	-
17	0	+	+	+	-	-	-	-
18	1	+	-	+	+	+	+	+
19	1	+	+	+	+	+	+	+
20	1	-	+	+	+	+	+	+
The efficiency measure μ		0.75	0.8	0.85	0.7	0.9	0.75	0.7

Assuming that T is the training set, S is the testing set, $X = \{x_1, x_2, \ldots, x_n\}$ is the group of classifier, F is a basic fusion operator, and A is a training algorithm for the two-class problem. Then:

1) determine the efficiency measure based on T, F and X according to Section V-A;
2) determine the integrand by solving the optimization problem (6) given in Section V-B;
3) according to the integrand obtained in (2), determine how many examples in S are assigned to different classifiers and their combinations;
4) train oracles by using algorithm A according to the second paragraph in Section V-C;
5) according to the oracles trained in (4), determine which examples in S are assigned to which combinations of classifiers;
6) let the combinations of classifiers, which are reflected in the formula of the upper integral, classify the assigned examples based on F;
7) calculate the final classification results.

It is worth noting that before the seven steps, the base classifiers are assumed to be known in advance. Moreover, it is worth noting that, regarding step (3) of the training oracles, we only need to train the oracles for those classifier combinations that appear in the formula of upper integral.

D. Illustration

Consider three classifiers $X = \{x_1, x_2, x_3\}$ and a dataset with 20 training examples shown in Table I, where the left part gives the dataset and the right part indicates the classification of the three individual classifiers and their four combinations on this dataset. The correct rate of classification for each of the seven cases is shown in the last row of Table I, which is considered the efficiency measure.

Solving the following linear programming problem:

$$\begin{aligned} \text{Maximum} \quad & (U)\int \{y_1, y_2, y_3\} d\mu \\ \text{Subject to} \quad & y_1 = a_1 + a_3 + a_5 + a_7 \le 0.75 \\ & y_2 = a_2 + a_3 + a_6 + a_7 \le 0.8 \\ & y_3 = a_4 + a5 + a_6 + a_7 \le 0.7 \\ & \sum_{i=1}^{7} a_i = 1 \\ & a_i \ge 0, i = 1, 2, \ldots, 7. \end{aligned} \tag{7}$$

We can obtain the integrand $f = \{y_1, y_2, y_3\} = \{0.75, 0.3, 0.7\}$. It implies that $[a_1, a_2, \ldots, a_7] = [0, 0.25, 0, 0.05, 0.7, 0, 0]$ which indicates that we may assign $0.25 \times 20 = 5$ examples to $\{x_2\}$, $0.05 \times 20 = 1$ example to $\{x_1, x_2\}$, and $0.7 \times 20 = 14$ examples to $\{x_1, x_3\}$, respectively.

From the solution of (7) we need to train three oracles for $\{x_2\}$, $\{x_1, x_2\}$ and $\{x_1, x_3\}$, respectively. For example, we consider to train an oracle for $\{x_1, x_2\}$. The training process of oracles for other classifier combinations is similar to that for $\{x_1, x_2\}$. The training set can be obtained from Table I, where examples correctly classified by $\{x_1, x_2\}$ are considered positive ones; otherwise, examples are considered negative ones. Based on this training set, we have trained a decision tree as the oracle for $\{x_1, x_2\}$. The allocation of examples in the test dataset is based on the trained oracle. Specifically, the trained oracle will predict whether it is appropriate for an example in the test dataset to be allocated to the classifier combination. The oracle trained for $\{x_1, x_2\}$ can correctly predict examples $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20\}$. The oracle cannot correctly predict the allocation of examples

TABLE II
ORACLE (A) ASSIGNS EXAMPLES TO CLASSIFIER A

Oracle (A) assigns examples to classifier A	
On the Testing set	On the Training set
A= $\{x_2\}$: 31,34,36,37,39	A= $\{x_2\}$: 1,2,3,8,9
A=$\{x_1, x_2\}$: 23	A=$\{x_1, x_2\}$: 4
A=$\{x_1, x_3\}$: 21,22,24,25,26,27,28,29,30,32,33,35,38,40	A =$\{x_1, x_3\}$: 5,6,7,10,11,12,13,14,15,16,17,18,19,20

TABLE III
TESTING SET

Example No.	Class	Predicted class used classifier	Used classifier and their combinations		
		$\{x_2\}$	$\{x_1, x_2\}$	$\{x_1, x_3\}$	
21	0	0 / $\{x_1, x_3\}$	1	1	0
22	1	1/ $\{x_1, x_3\}$	1	1	1
23	1	1/ $\{x_1, x_2\}$	1	1	0
24	0	0 / $\{x_1, x_3\}$	0	0	0
25	1	1 / $\{x_1, x_3\}$	0	0	1
26	1	1 / $\{x_1, x_3\}$	1	1	1
27	1	1 / $\{x_1, x_3\}$	1	1	1
28	0	0 / $\{x_1, x_3\}$	1	0	0
29	0	0 / $\{x_1, x_3\}$	0	1	0
30	1	1 / $\{x_1, x_3\}$	1	1	1
31	1	0 / $\{x_2\}$	0	1	1
32	0	0 /$\{x_1, x_3\}$	0	0	0
33	0	0 / $\{x_1, x_3\}$	1	0	1
34	1	1 / $\{x_2\}$	1	1	0
35	1	1 /$\{x_1, x_3\}$	1	1	1
36	0	0/ $\{x_2\}$	0	0	1
37	0	0 /$\{x_2\}$	0	0	0
38	1	1 / $\{x_1, x_3\}$	1	1	1
39	1	1 / $\{x_2\}$	1	1	1
40	0	0/ $\{x_1, x_3\}$	0	0	0
The correct rate		0.95	0.75	0.85	0.85

15 and 16. In other words, the oracle predicts that $\{x_1, x_2\}$ can correctly classify examples 15 and 16, but actually, $\{x_1, x_2\}$ cannot. It indicates that the prediction accuracy of the oracle trained for $\{x_1, x_2\}$ is 0.9. Table II shows the working status of three oracles for $\{x_1, x_2\}$, $\{x_1, x_3\}$ and $\{x_2\}$, respectively. By analyzing Table II, it can be seen that instances 15 and 16 were correctly classified by the combination $\{x_1, x_3\}$, not by $\{x_1, x_2\}$.

The example assignment schedule in the right of Table II leads to a correct classification rate of 0.95 on the training set. Furthermore, applying the three oracles to the testing set (Table III), we have the example assignment schedule shown in the left of Table II and the predicted results shown in Table III that indicates a 0.95 correct rate on the testing set.

There is a need to clearly point out the following two key points: 1) the value of integral does not represent the correct rate of classification but only represents a type of classification efficiency since the oracles do not randomly but do selectively assign examples to their corresponding classifiers; and 2) the final correct rate of classification is bigger than the correct rate of best combination $\{x_1, x_3\}$ on both the training set and the testing set. This illustration indicates that the performance of the upper integral can be higher than that of any of the combinations if the oracles work well.

E. Characteristics of the Model

The upper integral is used in our model to maximize the classification efficiency by reasonably assigning unseen examples to classifier combinations. It is different from the existing classifier fusion system because the upper integral is not the fusion operator but there is a basic fusion operator for evaluating the efficiency measure in our model. Fig. 2 shows the difference between our proposed model and the general fusion model such as bagging and the existing fuzzy integral based models. Fig. 2 explicitly indicates the difference between our model and the existing fuzzy integral based models of fusion, which is located in the third step of Fig. 2. In the existing fuzzy integrals based model, fuzzy integrals are just fusion operators in the third step, while in our proposed model, the upper integral is used to form optimization problem (6) (instead of being a fusion operator) in the third step. Another significant difference between (a) and (b) in Fig. 2 is that all classifiers are used to classify all unseen examples in the general fusion model, while only some of classifier combinations are used to classify assigned examples in our model.

If there are n classifiers, there will be $2^n - 1$ parameters in the optimization problem (6). When the number of classifiers increases, there is an explosion in the number of parameters

TABLE IV
DATASETS USED IN OUR EXPERIMENTS

Dataset	Size	Number of classes	Number of attributes	dataset	Size	Number of classes	Number of attributes
Iris	150	3	5	Clouds	5000	2	3
Pima	768	2	9	Concentric	2500	2	3
Breast Cancer	683	2	10	SVMguide1	7089	2	5
Ionosphere	351	2	35	Letter	20 000	26	17
Heart	270	2	10	Waveform	5000	3	22
Credit	666	2	7	Waveform+noise	5000	3	41
Abalone	4177	29	8	MAGIC04	19 020	2	11

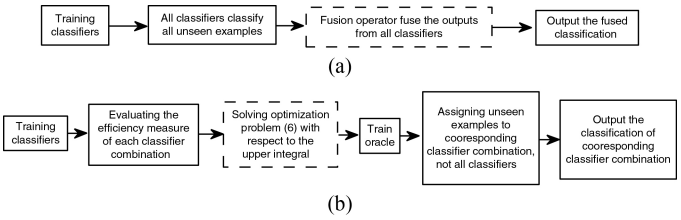


Fig. 2. Difference of bagging and upper integral model. (a) General fusion model. (b) Fusion model based on upper integral.

to be solved by the optimization problem. It is difficult to find the best strategy for solving the explosion problem. When the number of base classifiers is small, we can consider all the combinations. When it is large, we have two strategies to tackle the parameter explosion problem. One is to limit the number of classifiers to be combined, and the other is to randomly select a certain number of combinations without limiting the number of classifiers to be combined. Experiments show that the former is easy to implement and the latter depends largely on the selected combinations.

VI. EXPERIMENT RESULTS

In order to know how well the upper integral-based fusion model, an empirical study is performed in this section. Fourteen datasets are respectively selected from the UCI machine learning repository [44]; LIBSVM available at http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/; and ELENA datasets are collected via anonymous ftp: ftp.dice.ucl.ac.be. All the referred datasets have been extensively used in testing the performance of different classifiers. They are sized from 150 to 20 000, and the detailed information is summarized in Table IV. In Section VI-A, the upper integral fusion model is compared with bagging [35] and boosting [36] with different number of base classifiers on concentric dataset. Bagging and boosting as metalearning algorithms are widely used [47], [48]. In Section VI-B, three types of base classifiers, i.e., fuzzy decision trees, neural networks, and least-squares support vector machines, are, respectively, implemented in experimental comparisons.

1) Fuzzy decision trees [12]. The fuzzy decision tree induction process consists of four steps: 1) fuzzifying the training data; 2) inducing a fuzzy decision tree; 3) converting the decision tree into a set of rules; and 4) applying fuzzy rules for classification. In fuzzifying data, we use the triangular membership function and select its slopes in the way that adjacent membership functions cross at the membership value 0.5. The induction process consists of the following steps.
   a) *Step* 1: Select the attribute with the smallest classification ambiguity as the root decision node.
   b) *Step* 2: Delete all empty branches of the decision tree node. If the truth level of classifying into one class is above a given threshold β, terminate the branch as a leaf. Otherwise, investigate whether an additional attribute will further partition the branch and further reduce the classification ambiguity. If yes, select the attribute with the smallest classification ambiguity as a new decision node from the branch. If not, terminate this branch as a leaf. At the leaf, all examples will be labeled to one class with the highest truth level.
   c) *Step* 3: Repeat step 2 for all newly generated decision nodes until no further growth is possible, the decision tree then is completed.

   In the inducing processes of fuzzy decision tree there are two important parameters: the significant level α and the truth level threshold β. An example belongs to a branch only when the corresponding membership is greater than α. The parameter α plays a very crucial role in filtering insignificant evidences, therefore eliminating insignificant branches and leaves. The truth level threshold β controls the growth of the tree. In our experiments, the truth level threshold and the significant level are empirically selected as $\beta = 0.8$ and $\alpha = 0.4$, respectively. More detailed information could be found from [12].
2) Neural networks. We use the MATLAB neural network toolbox to train the back-propagation neural networks (BP-NN) as our base classifiers, where the number of hidden layers is set as 1 and the transfer function is fixed as hyperbolic tangent sigmoid. The number of hidden neurons used in each dataset, which is dependent of problems and is empirically given, is listed Table V.
3) Support vector machine (SVM). Support vector machines are powerful methodologies for solving problems in nonlinear classification with convex optimization problems. Least squares support vector machines (LS-SVM) are reformulations to the standard SVMs which solves linear KKT systems instead of a convex quadratic programming problem.

TABLE V
NUMBER OF HIDDEN NEURONS FOR EACH DATASET USED IN OUR EXPERIMENTS

Dataset	Iris	Pima	Breast Cancer	Heart	Ionosphere	Credit	Abalone
Number of Hidden Neurons	5	8	10	8	11	13	40
dataset	Clouds	Concentric	SVMguide1	Letter	Waveform	Waveformnoise	MAGIC04
Number of Hidden Neurons	9	7	20	80	20	30	50

We use the MATLAB LS-SVMlab Toolbox available at http://www.esat.kuleuven.ac.be/sista/lssvmlab/ to train the least-squares support vector machines with RBF kernel as our base classifier. There are two parameters, i.e., γ and σ. γ is the regularization parameter and σ is the kernel function parameter. For γ low minimizing of the complexity of the model is emphasized, and for γ high good fitting of the training data points is stressed. A large σ indicates a stronger smoothing. For each dataset, we estimate the generalization accuracy using different kernel parameter σ and regularization parameter γ with $\sigma = [2^4, 2^3, 2^2, \ldots, 2^{-10}]$ and $\gamma = [2^{12}, 2^{11}, 2^{10}, \ldots, 2^{-2}]$. We conduct a ten-fold cross validation on each dataset and get the LS-SVM by selecting the pair of (σ, γ), which achieves the best average cross-validation accuracy. The SVM is suitable for the two-class problem. For multiple-class problems in our paper, we use the $1-$against-1 strategy, a common mechanism for transferring more-than-two class to two-class problems.

In our experiment, in order to guarantee a fair comparison with bagging/boosting, both the type and the training algorithm of oracles are selected to be exactly the same as that of the base classifiers. Furthermore, the parameter selected in training oracles is also the same as that in the base classifier training.

A. Performance on Concentric Dataset

The concentric dataset are 2-D with two classes and uniform concentric circular distribution. The points in one class are uniformly distributed into a circle of radius 0.3 centered on $(0.5, 0.5)$. The points in another class are uniformly distributed into a ring centered on $(0.5, 0.5)$ with internal and external radius equal to 0.3 and 0.5, respectively. Since there exists a classification boundary that can completely separate one class from the other, the theoretical classification accuracy can attain to 100% (if the boundary is learned). The graphical representation of the concentric data is given in Fig. 3.

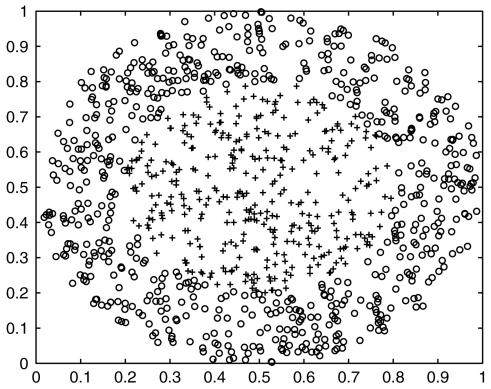


Fig. 3. Concentric data.

Here, the base classifier is a fuzzy decision tree. The basic fusion operators are the default ones in bagging and boosting. That is to say, the fusion operator in bagging is majority vote, and the one in boosting is the weighted majority vote. In the upper integral model, examples are assigned to different base classifiers or their combinations for classification. When an example is assigned to a combination, the final classification result is obtained through fusing the outputs of base classifiers in a combination. The fusion strategy is weighted majority vote and the weights for these base classifiers are the same as in boosting. The proposed algorithm and boosting are implemented with different number of base classifiers, while both of them adopt weighted majority vote as the fusion strategy. When base classifiers are assumed to be generated by bagging, the majority vote was used as the fusion strategy for both of the proposed algorithm and bagging.

First, we examine the performance of the upper integral model with different amount of base classifiers and their all possible combination on concentric data. The ten-fold cross validation is performed. Figs. 4 and 5 show the change of performance with the number of base classifiers. The classification accuracy increases slowly with amount of base classifiers. The accuracy of the upper integral model is higher than those of bagging and boosting. The accuracy of boosting is higher than that of bagging. Moreover, from Figs. 4 and 5, it can be observed that the performance of upper integral is slightly different from that of bagging, while its difference with boosting is more obvious. Maybe, this is due to the fact that, in boosting, the classifiers are obtained sequentially, in contrast to bagging where classifiers are obtained randomly and independently of the previous step of the algorithm. On concentric dataset, it is investigated that the interaction among classifiers in boosting is stronger than that in bagging and the upper integral could well capture this interaction. We use the interaction index [49] to show the interaction among classifiers in bagging and boosting on concentric dataset. The efficiency measure μ determined on training data shows the performance of several classifiers jointly used at the same time. Thus, the interaction index of a classifier combination can be expressed as the interaction among base classifiers. The definition of interaction index of a combination A is

$$I(A) = \sum_{B \supset A} \frac{1}{|B \backslash A| + 1} \mu(B).$$

The average interaction index for combinations is listed in Table VI that shows that the interaction in boosting is stronger than that in bagging.

TABLE VI
AVERAGE INTERACTION INDEX OF CLASSIFIER COMBINATION FOR BAGGING AND BOOSTING RESPECTIVELY ON CONCENTRIC DATASET

Number classifiers in combination	1	2	3	4	5	6	7	8	9
Bagging	64.0388	36.7587	20.5576	11.7308	6.7950	4.0091	2.4332	1.5192	1.1097
Boosting	66.7618	38.8224	21.7974	11.8444	6.8542	4.1617	2.4660	1.5333	1.1164

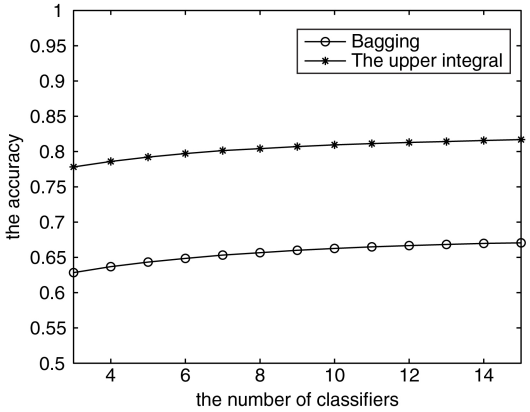


Fig. 4. Comparison between upper integral model and bagging on concentric data.

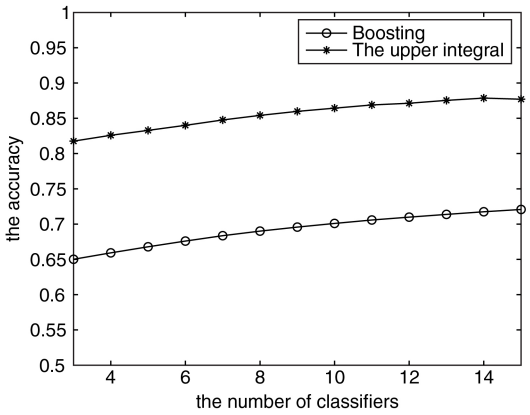


Fig. 5. Comparison between upper integral model and boosting on concentric data.

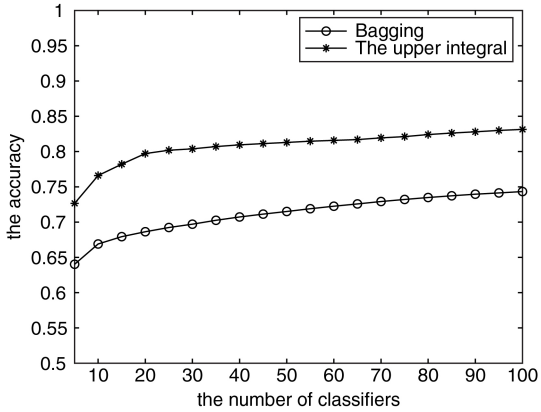


Fig. 6. Comparison between bagging and upper integral model with potential 2.

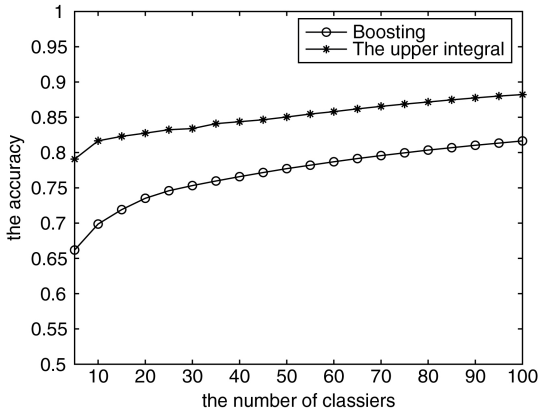


Fig. 7. Comparison between boosting and upper integral model with potential 2.

Second, we examine the performance of the upper integral with different amount of base classifiers and their partial combinations (not all possible combinations). This is because, when the number of base classifiers is large, the parameters in efficiency measure increase exponentially. The maximum number of classifiers in a combination to be combined is here called the potential. When the potential is one, we only consider individual classifiers and do not consider their any combination. A small potential can dramatically reduce the number of parameters in efficiency measure, but it means to loss some accuracy. Practically, we need to acquire a balance between the potential and accuracy. Figs. 6 and 7 show the performance with potential being 2 on concentric dataset. Because of the restricted potential, the accuracy of the upper integral is lower than that in Figs. 4 and 5 with the same amount of base classifiers. It is noted that the accuracy increases with the amount of base classifiers and the difference between the upper integral and bagging/boosting is significant. Figs. 6 and 7 show that in most cases the upper integral model could improve the classification performance with limited potential.

It is noted the accuracy is much lower than the theoretical accuracy 100%. A reason is that the performance depends on both the type of base classifiers and the parameters used in the base classifiers. When the base classifiers (fuzzy decision trees) are replaced with neural networks, the accuracy can be much improved (Table IX).

B. Comparison With Different Types of Base Classifiers

In the following experiment, we will continue examining the upper integral model on the other datasets. We perform ten-fold cross validation 20 times on each dataset. Both bagging and boosting contain 100 base classifiers always.

First, we make some comparisons between bagging/boosting and the upper integral model in which only single

TABLE VII

COMPARISON BETWEEN UPPER INTEGRAL MODEL AND BAGGING/BOOSTING ON 14 DATASETS (AVERAGE TESTING ACCURACY AND WILCOXON SIGNED-RANKS TEST)

Dataset	Upper Integral based bagging	Bagging	Wilcoxon signed-ranks test against bagging	Upper Integral based boosting	Boosting	Wilcoxon signed-ranks test against boosting
Iris	0.9715	0.9778	−0.0014	0.9716	0.9717	−0.0018
Pima	0.7984	0.7573	−3.164	0.7861	0.7613	−2.149
Breast Cancer	0.9487	0.9241	−2.471	0.9551	0.9181	−2.649
Ionosphere	0.7736	0.7311	−3.179	0.7624	0.7231	−3.201
Heart	0.7479	0.7232	−2.106	0.7472	0.7277	−2.0103
Credit	0.8386	0.7986	−4.2768	0.8577	0.8092	−4.271
Abalone	0.2147	0.2131	−1.3189	0.2178	0.1968	−2.101
Clouds	0.8561	0.8402	−3.0194	0.8799	0.8573	−2.2016
Concentric	0.834	0.7489	−5.001	0.8882	0.8202	−5.1632
SVMguide1	0.7678	0.7301	−3.3183	0.7715	0.7374	−4.1138
Letter	0.8916	0.8784	−2.017	0.9182	0.9019	−0.0041
Waveform	0.8403	0.8289	−2.403	0.8317	0.8231	−1.979
Waveform+noise	0.8292	0.8171	−2.014	0.8217	0.8024	−2.106
MAGIC04	0.7856	0.7603	−2.2792	0.7871	0.7697	−2.317

TABLE VIII

AVERAGE TESTING ACCURACY, STANDARD DEVIATION AND TIMES (SECONDS) ON 14 DATASETS USING FUZZY DECISION TREE, AND TWO BASIC FUSION OPERATORS: MEAN AND CHOQUET INTEGRAL, NO VALIDATION SET

Data	Best classifier		Average classifiers		Basic fusion	Upper integral			Bagging			Boosting		
set	Mean	Std Dev	Mean	Std Dev	operator	Mean	Std Dev	Time	Mean	Std Dev	Time	Mean	Std Dev	Time
Iris	0.9587	0.0286	0.9272	0.0129	Average	0.9723	0.0114	15.012	0.9718	0.0097	73.0417	0.9719	0.011	69.3219
					Choquet	0.9720	0.0112	4223.9						
Pima	0.7584	0.0247	0.7248	0.0194	Average	0.7847	0.0118	142.382	0.7768	0.0132	885.078	0.7801	0.0101	855.192
					Choquet	0.7862	0.0121	2808.8						
Breast Cancer	0.9372	0.0219	0.9107	0.0115	Average	0.9544	0.0114	132.6	0.9488	0.0169	611.4	0.9461	0.0121	619.8
					Choquet	0.9635	0.0128	2722.3						
Ionosphere	0.7571	0.0306	0.7089	0.0207	Average	0.7608	0.0162	148.2	0.7392	0.0143	732.6	0.7301	0.0175	722.4
					Choquet	0.7628	0.0173	2694.5						
Heart	0.7166	0.0472	0.6793	0.0328	Average	0.7524	0.0319	173.8	0.7293	0.0294	698.6	0.7264	0.0227	712.8
					Choquet	0.7647	0.0313	2664.2						
Credit	0.7549	0.0327	0.6887	0.0289	Average	0.8147	0.0114	201.4	0.8035	0.0217	821.8	0.8083	0.0182	830.7
					Choquet	0.8225	0.0163	2759.4						
Abalone	0.1979	0.0019	0.1781	0.0008	Average	0.2103	0.0012	928.6	0.2114	0.0009	3327.2	0.2074	0.0012	3274.7
					Choquet	0.2213	0.0018	6992.7						
Clouds	0.8536	0.0217	0.8146	0.0117	Average	0.8549	0.0113	58.4	0.8479	0.0111	227.9	0.8545	0.0091	216.7
					Choquet	0.8674	0.0028	2685.8						
Concentric	0.8003	0.0174	0.7684	0.0063	Average	0.8738	0.0062	88.3	0.8450	0.0095	304.6	0.8316	0.0074	316.4
					Choquet	0.8794	0.0128	2727.1						
SVMguide1	0.7328	0.0171	0.7086	0.0201	Average	0.7613	0.0158	135.2	0.7401	0.0083	661.7	0.7433	0.0091	648.7
					Choquet	0.7814	0.0108	2704.3						
Letter	0.8594	0.0321	0.8253	0.0248	Average	0.8967	0.0137	2408.4	0.8791	0.0169	9175.4	0.8937	0.0138	9013.6
					Choquet	0.8973	0.0152	70744						
Waveform	0.8217	0.0074	0.7927	0.0049	Average	0.8471	0.0114	587.8	0.8311	0.0024	2418.2	0.8314	0.0012	2672.5
					Choquet	0.8625	0.0029	6626.6						
Waveform+noise	0.8002	0.0132	0.7798	0.0099	Average	0.8136	0.0106	1226.8	0.8143	0.0093	3531.7	0.8121	0.0104	3275.9
					Choquet	0.8217	0.0114	7030.7						
MAGIC04	0.7816	0.0221	0.7522	0.0146	Average	0.7913	0.0114	1014.8	0.791	0.0137	4961.4	0.7714	0.0081	4874.6
					Choquet	0.8047	0.0103	5835.9						

classifier and combinations of two classifiers are considered. The results for bagging and boosting are listed in Table VII. Fuzzy decision trees are used as the base classifiers. For both the comparisons with bagging and boosting, the upper integral model uses the 100 base classifiers trained by them. The fusion operators in these two sets of comparisons are, respectively, fixed as majority vote and weighted majority vote, where for the weighted majority vote, the weight for each individual classifier is determined during the training of boosting. It is worth noting that the computational complexity of the upper integral model is higher than that of bagging and boosting since it involves more processes such as determining the efficiency measure, solving the optimization problem (6), and training oracles.

In Table VII, we use the Wilcoxon signed-ranks test [45], [46] to see whether there exists significant difference between the two referred methods. The Wilcoxon signed-ranks test is a safe nonparametric alternative of the paired t-test for statistical comparison of two methods [45].

Let the performances of two methods be $p_1, p_2, \ldots, p_m$ and $q_1, q_2, \ldots, q_m$, respectively, in paired experiments. The differences $d_i = p_i - q_i$ are ranked according to their absolute values; average ranks are assigned in case of ties. Let R^+ be the sum of ranks for $d_i > 0$, and R^- the sum of ranks for $d_i < 0$. Ranks of $d_i = 0$ are split evenly among the sums

$$R^+ = \sum_{d_i>0} rank(d_i) + 0.5 \sum_{d_i=0} rank(d_i)$$

TABLE IX

AVERAGE TESTING ACCURACY, STANDARD DEVIATION AND TIMES (SECONDS) ON 14 DATASETS USING NEURAL NETWORKS, AND TWO BASIC FUSION OPERATORS: MEAN AND CHOQUET INTEGRAL, NO VALIDATION SET

Data	Best classifier		Average classifiers		Basic fusion	Upper integral			Bagging			Boosting		
set	Mean	Std Dev	Mean	Std Dev	operator	Mean	Std Dev	Time	Mean	Std Dev	Time	Mean	Std Dev	Time
Iris	0.9628	0.0198	0.9357	0.0105	Average	0.9734	0.0081	12.27	0.9736	0.0071	57.0417	0.9758	0.039	57.589
					Choquet	0.9782	0.092	2321.6						
Pima	0.7597	0.0308	0.7015	0.0189	Average	0.7649	0.0127	22.850	0.7523	0.0106	110.001	0.7502	0.0095	105.103
					Choquet	0.7739	0.0074	2110.3						
Breast Cancer	0.9411	0.0127	0.9218	0.0079	Average	0.9687	0.0102	25.7	0.9613	0.0137	92.14	0.9583	0.0118	94.7
					Choquet	0.972	0.0049	2358.2						
Ionosphere	0.9127	0.0217	0.8752	0.0169	Average	0.9294	0.0162	15.04	0.9286	0.0058	134.79	0.9301	0.0063	135.1
					Choquet	0.9358	0.0118	2349.8						
Heart	0.7792	0.0328	0.7321	0.0301	Average	0.8127	0.0291	13.016	0.8051	0.0184	107.07	0.8107	0.0073	108.12
					Choquet	0.8316	0.0241	2454.8						
Credit	0.7689	0.0352	0.6915	0.0341	Average	0.8063	0.0278	36.041	0.8072	0.0168	120.094	0.8005	0.0136	133.8
					Choquet	0.8225	0.0163	2759.4						
Abalone	0.2012	0.0046	0.1871	0.0023	Average	0.2321	0.0026	1831.7	0.2251	0.0018	7516.3	0.2190	0.0015	7561.2
					Choquet	0.2411	0.0037	26939.2						
Clouds	0.8862	0.0059	0.8542	0.0046	Average	0.8973	0.0073	10.57	0.8954	0.0007	116.6	0.8918	0.0016	119.5
					Choquet	0.9041	0.0026	2575.2						
Concentric	0.9510	0.0068	0.9326	0.0015	Average	0.9801	0.0012	22.7	0.9763	0.0008	94.6	0.9735	0.0006	95.3
					Choquet	0.9876	0.0008	2635.7						
SVMguide1	0.9748	0.0071	0.9612	0.0049	Average	0.9796	0.0077	35.4	0.9763	0.0073	354.7	0.9782	0.0028	368.9
					Choquet	0.9815	0.0018	2418.0						
Letter	0.9128	0.0132	0.8759	0.0124	Average	0.9296	0.0111	60737	0.9308	0.0106	309217	0.9227	0.0134	394056
					Choquet	0.9374	0.0104	83278						
Waveform	0.8099	0.0094	0.7916	0.0103	Average	0.8252	0.0104	137.8	0.8227	0.0031	810.9	0.8211	0.0028	817.1
					Choquet	0.8295	0.0041	5271.9						
Waveform+noise	0.7819	0.0115	0.7518	0.0095	Average	0.8201	0.097	196.1	0.8155	0.0068	1301.3	0.8071	0.0073	1538.2
					Choquet	0.8215	0.0114	5921.0						
MAGIC04	0.8563	0.0169	0.0.8364	0.0153	Average	0.8809	0.0091	814.8	0.8709	0.0063	4584.3	0.8772	0.0081	4814.1
					Choquet	0.8947	0.0103	5138.5						

TABLE X

AVERAGE TESTING ACCURACY, STANDARD DEVIATION AND TIMES (SECONDS) ON 14 DATASETS USING LEAST-SQUARES SUPPORT VECTOR MACHINES, THE BASIC FUSION OPERATOR: MAJORITY VOTE, NO VALIDATION SET

Data	Best classifier		Average classifiers		Upper integral			Bagging			Boosting		
set	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Time	Mean	Std Dev	Time	Mean	Std Dev	Time
Iris	0.9637	0.0097	0.9436	0.0072	0.9837	0.0091	2.121	0.9789	0.0030	6.792	0.9806	0.0021	6.281
Pima	0.7801	0.0041	0.7726	0.0024	0.7994	0.0023	1.862	0.7864	0.0015	9.128	0.7857	0.0017	9.2051
Breast Cancer	0.9784	0.0068	0.9602	0.0057	0.9827	0.0039	1.672	0.9812	0.0026	7.8313	0.9813	0.0028	7.0384
Ionosphere	0.9295	0.0086	0.9144	0.0084	0.9528	0.0072	1.781	0.9487	0.0041	6.6629	0.9416	0.0037	6.2409
Heart	0.8077	0.0086	0.7891	0.0122	0.8269	0.0071	1.049	0.8123	0.0050	3.6287	0.8194	0.0038	3.5691
Credit	0.8268	0.0078	0.8066	0.0082	0.8435	0.0088	1.037	0.8359	0.0054	3.4818	0.8364	0.0037	3.5827
Abalone	0.2317	0.010	0.2194	0.0091	0.2458	0.094	4028.2	0.2388	0.0082	19621	0.2412	0.089	18746
Clouds	0.9303	0.0071	0.9187	0.0077	0.9423	0.093	39.71	0.9368	0.0082	190.29	0.9318	0.083	189.72
Concentric	0.9868	0.0030	0.9786	0.0009	0.9894	0.007	9.316	0.9879	0.0009	48.08	0.9895	0.0008	45.69
SVMguide1	0.9817	0.006	0.9806	0.009	0.9824	0.004	91.47	0.9822	0.003	492.60	0.9819	0.0008	481.59
Letter	0.9324	0.0388	0.9174	0.0274	0.9486	0.0214	41941	0.9392	0.0147	176192	0.9403	0.0172	130164
Waveform	0.8014	0.0235	0.7884	0.0194	0.8283	0.0187	108.81	0.8290	0.0128	623.01	0.8257	0.0181	629.81
Waveform+noise	0.8156	0.0251	0.8002	0.0117	0.8312	0.0148	181.29	0.8241	0.0170	861.07	0.8216	0.0162	850.63
MAGIC04	0.8831	0.0206	0.8671	0.0210	0.8884	0.020	1097.3	0.8892	0.0179	5622.63	0.8878	0.0182	4927.14

$$R^- = \sum_{d_i<0} rank(d_i) + 0.5 \sum_{d_i=0} rank(d_i).$$

Let T be the smaller one of the sums, $T = min(R^+, R^-)$. Most books on general statistics include a table of exact critical values for T for m up to 25. With an increasing value of m, the distribution of the statistics

$$z = \frac{T - \frac{1}{4}m(m)}{\sqrt{\frac{1}{24}m(m+1)(2m+1)}}$$

will approximate normal distribution. With the confidence level of $\alpha = 0.05$, the difference between the two referred methods will be treated as significantly different if the value of the Wilcoxon signed-ranks test z is smaller than -1.96.

In our experiment, ten-fold cross validation is repeated 20 times. Thus, with 20×10 values, so the Wilcoxon signed-ranks test is supposed to have normal distribution. It is easy to observe from Table VII that the proposed method is significantly different with bagging/boosting on 12 datasets out of 14. While on Iris dataset, the performance of the upper integral model is unsatisfactory, perhaps this is due to the fact

TABLE XI
WILCOXON SIGNED-RANKS TEST ON 14 DATASETS FOR TESTING SIGNIFICANCE OF DIFFERENCE BETWEEN UPPER INTEGRAL MODEL AND BAGGING/BOOSTING

Data	Basic fusion	Fuzzy decision tree		Neural network		LS-SVM	
set	operator	bagging	boosting	bagging	boosting	bagging	boosting
Iris	Average	−0.033	0.037	−0.029	0.062	−0.017	−0.047
	Choquet	0.042	−0.043	−0.039	0.071		
Pima	Average	−2.278	−2.019	−2.003	−2.036	−1.987	−2.137
	Choquet	−2.734	−2.179	-2.370	−2.396		
Breast Cancer	Average	−3.631	−3.017	−1.972	−1.989	−0.582	−0.568
	Choquet	−3.952	−3.516	−1.996	−2.005		
Ionosphere	Average	−4.017	−4.521	−0.048	0.028	−1.976	−2.160
	Choquet	−4.966	−4.985	−2.000	−0.416		
Heart	Average	−5.106	−5.277	−1.984	−1.648	−2.413	−2.102
	Choquet	−5.472	−5.852	−2.164	−2.087		
Credit	Average	−3.156	−3.007	0.025	−1.984	−1.979	−1.971
	Choquet	−3.739	−3.461	−2.572	−2.597		
Abalone	Average	−1.065	−0.971	−2.618	−3.028	−2.003	−1.528
	Choquet	−1.993	−2.058	−2.917	−3.301		
Clouds	Average	−1.984	−1.693	−1.237	−1.981	−1.832	−1.985
	Choquet	−2.318	−2.006	−1.974	−1.985		
Concentric	Average	−4.671	−4.864	−1.981	−1.582	−0.691	0.030
	Choquet	−4.597	−4.968	−1.977	−1.993		
SVMguide1	Average	−3.174	−3.108	−0.973	−0.826	−0.023	−0.307
	Choquet	−4.041	−3.846	−1.969	−1.973		
Letter	Average	−2.571	−0.877	0.27	−1.968	−2.319	−2.201
	Choquet	−2.619	−1.481	−1.982	−2.004		
Waveform	Average	−2.182	−2.066	−0.472	−0.602	−0.018	−0.392
	Choquet	−2.215	−2.111	−1.621	−1.973		
Waveform+noise	Average	−0.120	−0.795	−1.987	−2.033	−1.988	−1.994
	Choquet	−1.978	0.207	−1.968	−2.308		
MAGIC04	Average	0.072	−1.983	−1.995	−1.982	−0.058	−0.203
	Choquet	−1.974	−2.081	−2.616	−2.006		

that the performance of the base classifier is already close to the optimal. Experiments show that in most case the best classifier can be selected or can be contained in the selected combinations on Iris dataset. The Wilcoxon signed-rank test shows that the upper integral model can improve significantly the performance of bagging and boosting.

Although we consider the combination of potential not more than two, the upper integral still could significantly improve the classification performance. It indicates that the upper integral model could sufficiently model and handle the interaction among base classifiers. The interaction between only two classifiers could not be ignored, but the performance of oracles influences the upper integral model. The degrees of improvement produced by the upper integral are different. For instance, the performance of our model on concentric and Credit datasets is better than that on Pima and Heart datasets. We now try to analyze the underlying reason for the unsatisfactory performance on datasets Pima and Heart. It is found that in both these two datasets, there are a number of examples that have similar conditional attribute values but different decision attribute values. This statement implies an unclear boundary between the two classes, which leads to the poor performance of the algorithms.

In the following, we use ten base classifiers that are trained with randomly selected 80% attributes of each dataset. It is an efficient way to obtain diverse base classifiers by using different attributes [33]. Also, in the optimization problem (6) all combinations are considered. Three types of base classifiers are used: fuzzy decision trees, neural networks, and LS-SVMs. Majority vote is used for LS-SVMs in the upper integral model. Two fusion operators, average and Choquet integral, are used for fuzzy decisions and neural networks in the upper integral model. In Choquet fusion operator, the λ-measure [43] is used and the density of the λ-measure is defined [25]

$$g_i^j = \frac{p_i^j}{\sum_{k=1}^{n} p_k^j}$$

where g_i^j is the measure of classifier x_i for class j, p_i^j is the accuracy of classifier x_i for examples in class j.

The results are listed in Tables VIII–X. The upper integral model can improve the classification performance in most cases. When only ten base classifiers are used, the upper integral model can achieve similar performances with bagging/boosting by consuming much less time. When neural networks are used as the base classifiers, the performances of all the three methods (the upper integral model, bagging and boosting) are higher than those with fuzzy decision trees as base classifiers on most datasets. It implies that the final classification performance is dependent on the type of the base

TABLE XII
AVERAGE TESTING ACCURACY ON 14 DATASETS WHEN USING VALIDATION SET

Data set	Fuzzy decision tree		Neural network		LS-SVM
	Average	Choquet	Average	Choquet	
Iris	0.9712	0.9711	0.9765	0.9751	0.9786
Pima	0.7812	0.7855	0.7587	0.7688	0.7886
Breast Cancer	0.9531	0.9648	0.9661	0.9732	0.9792
Ionosphere	0.7516	0.7643	0.9128	0.9194	0.9474
Heart	0.7418	0.7521	0.8138	0.8294	0.8295
Credit	0.8194	0.8276	0.7987	0.8228	0.8374
Abalone	0.2082	0.2219	0.2206	0.2278	0.2492
Clouds	0.8598	0.8677	0.8987	0.9103	0.9482
Concentric	0.8862	0.8867	0.9831	0.9898	0.9904
SVMguide1	0.7713	0.7961	0.9786	0.9801	0.9816
Letter	0.8922	0.8914	0.9317	0.9389	0.9528
Waveform	0.8499	0.8658	0.8328	0.8374	0.8319
Waveform+noise	0.8226	0.8297	0.8075	0.8112	0.8251
MAGIC04	0.8026	0.8214	0.8857	0.8891	0.8972

TABLE XIII
ACCURACY, STANDARD DEVIATION OF C4.5

Dataset	Iris	Pima	Breast cancer	Ionosphere	Heart	Credit	Abalone
Accuracy	0.9519	0.7464	0.9513	0.8911	0.7832	0.7796	0.2072
Std Dev	0.0493	0.0461	0.0257	0.0477	0.0719	0.0483	0.009
Dataset	Clouds	Concentric	SVMguide1	Letter	Waveform	Waveform noise	MAGIC04
Accuracy	0.8855	0.9782	0.9708	0.8813	0.7658	0.7534	0.8511
Std Dev	0.0183	0.0141	0.006	0.0065	0.0188	0.0182	0.0074

classifiers. In addition, it can be seen from Tables VIII and IX that the upper integral model is very time consuming when the Choquet integral is used as the basic fusion operator. Moreover, the performance of the upper integral model, bagging and boosting, cannot be significantly higher than the performance of single SVM for some two-class problems. We think that the major reason is that the single has already had the very high accuracy on these datasets and then some fusion schemes, such as bagging, boosting, and our upper integral model, will not have a significant improvement of accuracy. But for most multiple-class problems, the performance of single SVM is not as good as the performance of the upper integral model, bagging and boosting.

A statistical test of accuracy difference between the upper integral model and bagging/boosting is conducted on the selected datasets. The results of the Wiloxon signed-ranks test are listed in Table XI. The testing results show that the difference is statistically significant over at least a half number of datasets (and the performance of the upper integral model is dependent on the type of base classifiers, the basic fusion operator, and the oracles). Details are listed as follows.

1) When fuzzy decision trees are the base classifiers and average is the basic fusion operator, the upper integral model significantly outperforms bagging on ten datasets and boosting on nine datasets, respectively (and the statistical testing results are not significant on the other datasets).
2) When fuzzy decision trees are the base classifiers and Choquet integral is the basic fusion operator, the upper integral model significantly outperforms bagging 13 datasets and boosting on 11 datasets (and the test is not significant on the other datasets).
3) When neural networks are the base classifiers and average is the basic fusion operator, the upper integral model significantly outperforms bagging on seven datasets and boosting on eight datasets, respectively (and the statistical testing results are not significant on the other datasets).
4) When neural networks are the base classifiers and Choquet integral is the basic fusion operator, the upper integral model significantly outperforms bagging/boosting on 12 datasets (and the statistical testing results are not significant on the other datasets).
5) When LS-SVMs are the base classifiers and majority vote is the basic fusion operator, the upper integral model significantly outperforms bagging/boosting on seven datasets, and on the other seven datasets there is no essential difference. An important reason for this improvement is that the oracle (which is also an LS-SVM classifier) becomes more precise with the parameter selection. In summary, the performance of the upper integral model is dependent on the type of base classifiers, the basic fusion operator, and the oracles.

An experiment for determining the efficiency measure is conducted based on an improved ten-fold cross validation procedure. Originally, the ten-fold cross validation takes nine folds as training set and the one remaining fold as the testing set. Now, one fold is used as the testing set, two folds are used

as the validation set, and the other seven folds as the training set to train the ten base classifiers. The result is listed in Table XII. It is shown that for large datasets, the performance is improved, while for small datasets, the difference is not obvious. The possible reason is that for large datasets, the validation set makes the efficiency measure more objective, while for small datasets, this impact is trivial. Generally, for large datasets, it is a good choice to avoid overfitting by determining the efficiency measure with validations. In the meantime, it indicates that by selecting suitable parameters via cross-validation, the accuracy of LS-SVM is significantly improved and is much higher than those of fuzzy decision tree and neural network on 13 datasets.

The accuracy of single classifier C4.5 (20 times ten-fold cross validation) is listed in Table XIII. Compared with Table XII, the experimental results show that the average accuracy of the upper integral model is higher than that of C4.5 on eight datasets if the base classifiers are fuzzy decision trees and on 14 datasets if the base classifiers are neural networks and LS-SVMs. It confirms that the performance of the upper integral model (which is dependent on the type of base classifiers and the oracles) is generally better than the single decision tree-like classifier such as C4.5 in which the parameters have been optimized.

VII. Conclusion

This paper proposed a multiple classifier fusion method based on the upper integral to most effectively use the individual classifiers and their combinations. The difficulty of determining the fuzzy measures was avoided by regarding the accuracies of classifier combinations as an efficiency measure defined on the power set of classifier set. The upper integral was used to determine the proportions of examples 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 was conducted. Theoretically, the definition of upper integrals indicated that the accuracy of upper integral based fusion system was not lower than that of any individual base classifier. Practically, it may not be true. The reason is that the classification depends on the oracles, which usually have training errors. Experimentally, in most cases, the accuracy of upper integral based fusion system was not lower than that of any individual base classifier, as well as any combination of the base classifiers. The Wilcoxon signed-ranks tests demonstrated that the improvement produced by the upper integral was significant.

Our future and ongoing works on this topic focus on the following three problems.

1) Simplifying the upper integral model including the efficiency measure such that it can be suitable classification problems with a large number of features. It involves the approximate representation of the upper integral model and the learning of structured efficiency measure.
2) Improving the upper integral model when the examples are appearing incrementally (in other words, how to efficiently use the order of examples appearance to establish an incremental upper integral model for classification problems).
3) Optimizing the process of using oracles to select specific examples which are submitted to the base classifiers based on the determined proportion.

References

[1] M. Sugeno, "Theory of fuzzy integrals and its applications," Doctoral Thesis, Tokyo Inst. Technol., Tokyo Kanagawa, Japan, 1974.

[2] S. Weber, "Z-Decomposable measures and integrals for Archimedean t-conorms," *J. Math. Anal. Applicat.*, vol. 101, no. 1, pp. 114–138, 1984.

[3] H. Tahani and J. M. Keller, "Information fusion in computer vision using fuzzy integral," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 3, pp. 733–741, May–Jun. 1990.

[4] G. Giacinto, F. Roli, and L. Didaci, "Fusion of multiple classifiers for intrusion detection in computer networks," *Pattern Recogn. Lett.*, vol. 24, no. 12, pp. 1795–1803, 2003.

[5] D. Wang, J. M. Keller, C. A. Carson, K. K. McAdoo-Edwards, and C. W. Bailey, "Use of fuzzy-logic-inspired features to improve bacterial recognition through classifier fusion," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 4, pp. 583–591, Aug. 1998.

[6] K.-C. Kwak and W. Pedrycz, "Face recognition using fuzzy integral and wavelet decomposition method," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1666–1675, Aug. 2004.

[7] M. Grabisch and M. Sugeno, "Multiattribute classification using fuzzy integral," in *Proc. 1st IEEE Int. Conf. Fuzzy Syst.*, Mar. 1992, pp. 47–54.

[8] X. Z. Wang, A. X. Chen, and H. M. Feng, "Upper integral network with extreme learning mechanism," *Neurocomputing*, vol. 74, no. 16, pp. 2520–2525, 2011.

[9] K. Xu, Z. Wang, P.-A. Heng, and K.-S. Leung, "Classification by nonlinear integral projections," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 2, pp. 187–201, Apr. 2003.

[10] F. Roli and J. Kittler, "Fusion of multiple classifiers," *Inf. Fusion, Guest Edtorial*, vol. 3, no. 4, p. 243, 2002.

[11] S.-B. Cho and J. H. Kim, "Multiple network fusion using logic," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 497–501, Mar. 1995.

[12] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets Syst.*, vol. 69, no. 2, pp. 125–139, 1995.

[13] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, Jan.–Jun. 1996.

[14] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[15] V. N. Vipnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer, 1998.

[16] K. M. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 2, pp. 359–366, 1989.

[17] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.

[18] E. Schmitt, V. Bombardier, and L. Wendling, "Improving fuzzy rule classifier by extracting suitable features from capacities with respect to the choquet integral," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1195–1206, Oct. 2008.

[19] T. Butko, A. Temko, C. Nadeu, and C. Canton-Ferrer, "Fusion of audio and video modalities for detection of acoustic events," in *Proc. Interspeech*, 2008, pp. 123–126.

[20] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ, USA: Wiley-Interscience, 2003.

[21] A. Verikas, A. Lipnichas, and K. Malmqvist, "Soft combination of neural classifiers: A comparative study," *Pattern Recognit. Lett.*, vol. 20, no. 4, pp. 429–444, 1999.

[22] I. Bloch, "Some aspects of Dempster-Shafer evidence theory for classification of multimodality medical images taking partial volume effect into account," *Pattern Recognit.*, vol. 17, no. 8, pp. 905–919, 1996.

[23] T. Denceux, "A k-nearest neighbor classification rule based on Dempster-Shafer theory," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 5, pp. 804–813, May 1995.

[24] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton Univ. Press, 1976.

[25] J. M. Keller, P. Gader, H. Tahani, J. Chiang, and M. Mohamed, "Advances in fuzzy integration for pattern recognition," *Fuzzy Sets Syst.*, vol. 65, no. 3, pp. 273–283, 1994.
[26] D. S. Yeung, X. Wang, and E. C. Tsang, "Handling interaction in fuzzy production rule reasoning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 1979–1987, Oct. 2004.
[27] M. Grabisch and J. Nicolas, "Classification by fuzzy integral: Performance and tests," *Fuzzy Sets Syst.*, vol. 65, no. 2–3, pp. 255–271, 1994.
[28] Z. Wang, K.-S. Leung, and J. Wang, "A genetic algorithm for determining nonadditive set functions in information fusion," *Fuzzy Sets Syst.*, vol. 102, no. 3, pp. 463–469, 1999.
[29] R. Yang, Z. Wang, P.-A. Heng, and K.-S. Leung, "Fuzzified Choquet integral with a fuzzy-valued integrand and its application on temperature prediction," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 367–380, Apr. 2008.
[30] J. Wang and Z. Wang, "Using neural networks to determine Sugeno measures by statistics," *Neural Netw.*, vol. 10, no. 1, pp. 183–197, 1997.
[31] Z. Wang, K.-S. Leung, and G. J. Klir, "Apllying fuzzy measures and nonlinear integrals in data mining," *Fuzzy Sets Syst.*, vol. 156, no. 3, pp. 371–380, 2005.
[32] Z. Wang, W. Li, and K.-S. Leung, "Lower integrals and upper integrals with respect to nonadditive set functions," *Fuzzy Sets Syst.*, vol. 159, no. 6, pp. 646–660, 2008.
[33] K. Anil, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.
[34] M. Grabisch, "The application of fuzzy integrals in multicriteria decision making," *Eur. J. Oper. Res.*, vol. 89, no. 3, pp. 445–456, 1996.
[35] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[36] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
[37] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.
[38] T. Windeatt, "Diversity measures for multiple classifier system analysis and design," *Inform. Fusion*, vol. 6, no. 1, pp. 21–36, 2005.
[39] M. Grabisch, I. Kojadinovic, and P. Meyer, "A review of methods for capacity identification in Choquet integral based multiattribute utility theory applications of the Kappalab R package," *Eur. J. Oper. Res.*, vol. 186, no. 2, pp. 766–785, 2008.
[40] E. Takahagi, "A fuzzy measure identification method by diamond pairwise comparisons and phi(s) transformation," *Fuzzy Optimization Decision Making*, vol. 7, no. 3, pp. 219–232, 2008.
[41] Y.-C. Hu, "Pattern classification by multilayer perceptron using fuzzy integral-based activation function," *Appl. Soft Comput.*, vol. 10, no. 3, pp. 813–819, 2010.
[42] Z. Wang, K.-S. Leung, M.-L. Wong, and J. Fang, "A new type of nonlinear integrals and the computational algorithm," *Fuzzy Sets Syst.*, vol. 112, no. 2, pp. 223–231, 2000.
[43] Z. Wang and G. J. Klir, *Fuzzy Measure Theory*. New York, NY, USA: Plenum, 1992.
[44] K. Bache, M. Lichman, *UCI Repository of Machine Learning Databases and Domain Theories*. Irvine, CA, USA: Univ. California, 2013 [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html
[45] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.
[46] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
[47] R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares, "Using meta-learning to support data mining," *Int. J. Comput. Sci. Applicat.*, vol. 1, no. 1, pp. 31–45, 2004.
[48] L. Todorovski and S. Dzeroski, "Combining classifiers with meta decision trees," *Mach. Learn.*, vol. 50, no. 3, pp. 223–249, 2003.
[49] D. Denneberg and M. Grabisch, "Interaction transform of set functions over a finite set," *Inform. Sci.*, vol. 121, no. 1–2, pp. 149–170, 1999.

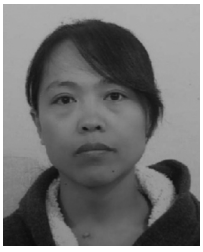
Xi-Zhao Wang (M'03–SM'04–F'12) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1998.

He is currently the Dean and Professor of the College of Mathematics and Computer Science, Hebei University, Hebei, China. From September 1998 to September 2001, he served as a Research Fellow in the Department of Computing, Hong Kong Polytechnic University, Hong Kong. He became a Full Professor and Dean of the College of Mathematics and Computer Science, Hebei University, in October 2001. He has over 160 publications, including four books, seven book chapters, and over 90 journal papers in IEEE Transactions on PAMI/SMC/FS, Fuzzy Sets and Systems, Pattern Recognition, etc. His H-index is 18 (up to April 2013). His current research interests include learning from examples with fuzzy representation, fuzzy measures and integrals, neuro-fuzzy systems and genetic algorithms, feature extraction, multiclassifier fusion, and applications of machine learning.

Dr. Wang has been the PI/Co-PI for 16 research projects supported partially by the National Natural Science Foundation of China and the Research Grant Committee of Hong Kong Government. He is a Senior Member of the IEEE (Board of Governor member in 2005, 2007-2009); the Chair of IEEE SMC Technical Committee on Computational Intelligence, an Associate Editor of IEEE TRANSACTIONS ON SMC, PART B; an Associate Editor of Pattern Recognition and Artificial Intelligence; a member of editorial board of Information Sciences; and an Executive Member of the Chinese Association of Artificial Intelligence. He was the recipient of the IEEE-SMCS Outstanding Contribution Award in 2004 and the recipient of IEEE-SMCS Best Associate Editor Award in 2006. He is the general Co-Chair of the 2002-2009 International Conferences on Machine Learning and Cybernetics, cosponsored by IEEE SMCS. He is a distinguished lecturer of IEEE SMC Society.

Ran Wang (S'09) received the Bachelors degree from the College of Information Science and Technology, Beijing Forestry University, Beijing, China, in 2009. She is currently pursuing the Ph.D. degree at the Department of Computer Science, City University of Hong Kong, Hong Kong.

Her current research interests include support vector machines, extreme learning machines, decision tree induction, active learning, multiclass classification, and the related applications of machine learning.



Hui-Min Feng received the B.Sc. and M.Sc. degrees in mathematics from Hebei University, Baoding, China, in 2002 and 2005, respectively.

Since 2005, she has been a Lecturer in the Faculty of Mathematics and Computer Science, Hebei University. Her current research interests include multiple classifier fusion system, feature extraction, fuzzy integral, and applications of machine learning.

Hua-Chao Wang received the Masters degree from the College of Mathematics and Computer Science, Hebei University, Baoding, China, in 2011.

He is currently a Researcher at the National Astronomical Observatories, Chinese Academy of Sciences, Beijing, China. His current research interests include rough sets, multiclass classification, decision tree, support vector machines, and the related applications of machine learning.