Journal of Intelligent & Fuzzy Systems 29 (2015) 1185–1196
DOI:10.3233/IFS-151729
IOS Press

Fuzziness based sample categorization for classifier performance improvement

Xi-Zhao Wang[a,*], Rana Aamir Raza Ashfaq[a] and Ai-Min Fu[b]
[a]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China*
[b]*College of Science, China Agricultural University, Beijing, China*

Abstract. This paper investigates a relationship between the fuzziness of a classifier and the misclassification rate of the classifier on a group of samples. For a given trained classifier that outputs a membership vector, we demonstrate experimentally that samples with higher fuzziness outputted by the classifier mean a bigger risk of misclassification. We then propose a fuzziness category based divide-and-conquer strategy which separates the high-fuzziness samples from the low fuzziness samples. A particular technique is used to handle the high-fuzziness samples for promoting the classifier performance. The reasonability of the approach is theoretically explained and its effectiveness is experimentally demonstrated.

Keywords: Fuzziness, misclassification, generalization, boundary point, Divide-and-Conquer strategy

1. Introduction

Classification is to determine (or estimate) a target function F that maps each object to a class label y. The process of determining the target function is called learning which is usually completed through a training algorithm. One way of learning is to minimize some error between F and its estimate f (classifier) on training samples. Wu et al. in [1] list the top-10 learning algorithms in data mining.

Generalization (i.e., the correct rate of classification on unseen samples of a classifier) is the most important index of classifier evaluation because the ultimate goal of learning is to reduce the error on unseen samples. There are many ways to study the generalization. One way is via the generation of training and testing samples. This type of studies includes re-sampling methods [2, 3], leave-one-out cross-validation [4, 5], approaches to assuming a specific distribution of testing samples and correspondingly developing generalization error formulation [6–8], online learning models on samples coming from a dependent source of data [9], etc. Another way to study the generalization is the estimation of error bound, which is a popular theoretical technique. From references one can find the results of existing studies on the estimation of generalization error bounds. For instance, the discussion about the error bounds to overcome overfitting problems [10]; structural risk minimization to link the generalization to training sample errors and the classifier complexity [11, 12]; the performance analysis on classifier ensemble bounds [13, 14]; the biased generalization bound [15]; and the bounds on the false and truth positive rates [16].

An interesting issue for classifier's generalization is how the fuzziness of a classifier relating to its prediction accuracy. From literature, except for [17, 18], we have not found specific studies on generalization based on the fuzziness of classifier outputs. The works in [17, 18], however, do not analyze the relationship between misclassification rate and the fuzziness of the classifier outputs, and their proposed methods are limited only to rule-based systems.

*Corresponding author. Xi-Zhao Wang, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China. Tel./Fax: +86 13700326811; E-mail: xzwang@szu.edu.cn.

1064-1246/15/$35.00 © 2015 – IOS Press and the authors. All rights reserved

This paper makes an attempt to investigate the generalization from the angle of fuzziness. It associates the fuzziness outputted by a classifier on a group of samples with the misclassified rate of this group of samples. We focus on the improvement of generalization ability through discovering relationship between sample misclassification and fuzziness of classifier outputs. The fuzziness is defined on a fuzzy vector (i.e., a vector in which each component is a number between 0 and 1), and therefore, this study on generalization is suitable for any type of classifiers with fuzzy vector output [e.g., 19, 20].

For a given trained classifier that outputs a membership vector, we demonstrate experimentally that samples with higher fuzziness outputted by the classifier mean a bigger risk of misclassification. Then based on the fuzziness quantity of output vector, all samples are categorized into 3 classes, i.e., low, middle, and high fuzziness classes. The high-fuzziness samples are particularly processed in order to improve the performance of the classifier, to be exact, the performance on the group of samples with high fuzziness.

The paper is organized as follows. Section 2 briefly reviews the fuzziness defined on a fuzzy vector, introduces the fuzziness of a classifier, and analyzes the relationship between fuzziness and boundary samples. Section 3 experimentally demonstrates that samples with higher fuzziness outputted by the classifier mean a bigger risk of misclassification. Section 4 gives the approach to handling the high fuzziness samples, the analysis on the approach's rationality, and the experimental demonstration. Section 5 concludes this paper.

2. Fuzziness of classifier output

The term fuzziness was first proposed in [21] by Zadeh in his famous fuzzy set theory. It describes a kind of imprecision existing in events which cannot be defined exactly and cannot be characterized by sharply defined collection of points. Zadeh also generalized probability measure of an event to fuzzy event and suggested using entropy in information theory to interpret the uncertainty associated with a fuzzy event. Luca and Termini [22] considered fuzziness as a type of uncertainty described by fuzzy sets, and defined a quantitative measure of fuzziness by non-probabilistic entropy which is very similar to Shannon's information entropy. They clearly proposed three properties that fuzziness measure should meet with for the first time. The three properties indicate that the degree of fuzziness should attain its maximum when all the memberships are equal to each other, and attain its minimum when all memberships are equal to either 0 or 1. Furthermore in [23], Luca and Termini extended their definition of entropy on fuzzy sets into the case of fuzziness of L-fuzzy sets, where the entropy was no longer a numerical quantity but a column matrix or a vector. It is interesting to study the difference among the terms "fuzziness","ambiguity", "uncertainty", "indefiniteness", "imprecision", etc., which may cause confusion in many situations. Klir et al. [24, 25] stated that vagueness or fuzziness is different from ambiguity, and then proposed two cognitive uncertainty measures. In general, vagueness or fuzziness is describing the difficulty of making sharp or precise distinctions in the world. Ambiguity, on the other hand, is connected together with people's cognitive uncertainty, that is, situations with two or more alternatives such that the choice between them is uncertain.

We now consider fuzziness as a type of cognitive uncertainty which is coming from the transition of uncertainty from one linguistic term to another, where a linguistic term is a fuzzy set defined on a certain universe of discourse. Moreover a linguistic term can be considered as a value of a linguistic variable. For example, Temperature is a linguistic variable which can take the linguistic values, say, hot, cool, middle, which are fuzzy sets defined on an interval of real numbers.

A mapping from a space $X \to [0, 1]$ is called a fuzzy set. All fuzzy set defined on X is denoted by $\mathscr{F}(X)$. As stated in literature [26], the fuzziness of a fuzzy set can be measured by a function $E : \mathscr{F}(X) \to [0, +\infty)$ that satisfies the following axioms:

(a) $E(\mu) = 0$ if and only if μ is a crisp set,

(b) $E(\mu)$ gets its maximum if and only if

$$\mu(x) = 0.5 \ \forall x \in X$$

(c) If $\mu \leq_s \sigma$, then $E(\mu) \geq E(\sigma)$,

(d) $E(\mu) = E(\mu')$, where $\mu'(x) = 1 - \mu(x)$ for

$$\forall x \in X,$$

(e) $E(\mu \cup \sigma) + E(\mu \cap \sigma) = E(\mu) + E(\sigma)$.

Regarding the third axiom, the sharpened order $\leq_S$ is defined as [22]:

$$\begin{aligned} \mu \leq_S \sigma \Leftrightarrow \min(0.5, \mu(x)) \geq \min(0.5, \sigma(x)) \\ \& \max(0.5, \mu(x)) \leq \max(0.5, \sigma(x)) \end{aligned} \tag{1}$$

Definition 2.1. Let $B = \{\mu_1, \mu_2, \ldots, \mu_n\}$ be a fuzzy set. According to [22], the fuzziness of B can be defined as

$$E(B) = -\frac{1}{n}\sum_{i=1}^{n}(\mu_i \ \log \mu_i + (1-\mu_i) \ \log(1-\mu_i)). \tag{2}$$

In fact one can construct many equations similar to (2), for example when $n=2$ and B is normalized (i.e. $\mu_1 + \mu_2 = 1$), we have

$$E_1(B) = 1 - \mu_1^2 - (1-\mu_1)^2 \tag{3}$$

$$E_2(S) = \begin{cases} \frac{\mu_1}{1-\mu_1} & 0 \le \mu_1 \le 0.5 \\ \frac{1-\mu_1}{\mu_1} & 0.5 \le \mu_1 \le 1 \end{cases} \tag{4}$$

It is easy to verify that each of Equations (2, 2a, 2b) indeed satisfies the above-mentioned axioms (a)-(e). Although there are many specific forms of fuzziness satisfying the above axioms (a)–(e), the further study on fuzziness's impact on misclassification shows that the classification performance is not sensitive to the specific equation forms of fuzziness. We continue our discussion based on Equation (2).

The fuzziness of a fuzzy set defined by (2) attains its minimum when every element absolutely belongs to the fuzzy set or absolutely not, i.e., $\mu_i = 1$ or $\mu_i = 0$ for each $i\,(1 \le i \le n)$; the fuzziness attains its maximum when the membership degree of each element is equal to 0.5, i.e., $\mu_i = 0.5$ for every $i = 1, \ 2, \cdots, n$.

When $n=2$, the picture of Equation (2) is depicted in Fig. 1, where the minimum and maximum are clearly shown in the extremes and the middle.

We now connect the fuzziness of a fuzzy vector with a classifier output. It is well found that many classifiers have the output form of fuzzy vector in which each component corresponds to the membership degree of the testing object belonging to a class. This type of classifiers includes neural networks, support vector machine, fuzzy decision trees, and etc. There is a need to clarify that for this type of classifiers, such as neural networks, a simple transformation can transfer the initial output to a form of fuzzy vector if components of the initial output are not in [0, 1].

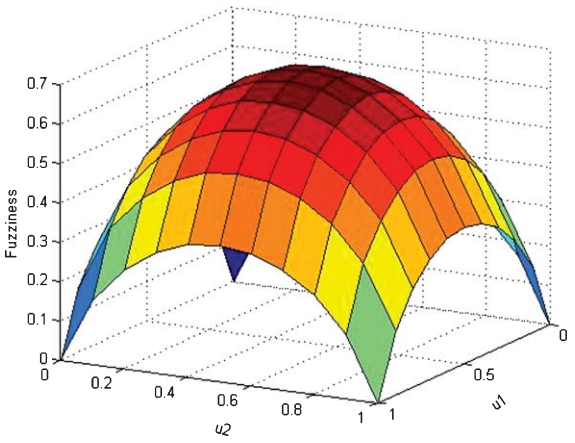

Fig. 1. Fuzziness of a two-dimensional vector.

Given a set of training samples $\{\mathbf{x}_i\}_{i=1}^{N}$, a fuzzy partition of these samples assigns the membership degrees of each sample to the c classes. The partition can be described by a membership matrix $\mathbf{U} = (\mu_{ij})_{c\times N}$, where $\mu_{ij} = \mu_i(\mathbf{x}_j)$ denotes the membership of the j-th sample $\mathbf{x}_j$ belonging to the i-th class. The elements in the membership matrix have to obey the following properties

$$\sum_{i=1}^{c}\mu_{ij} = 1, \ 0 < \sum_{j=1}^{N}\mu_{ij} < N, \ \mu_{ij} \in [0, \ 1] \tag{5}$$

Therefore, once the training procedure of a classifier completes, the membership matrix $\mathbf{U}$ upon the N training samples can be obtained. For the j-th sample $\mathbf{x}_j$, the trained classifier will give an output vector represented as a fuzzy set $\mu_j = (\mu_{1j}, \ \mu_{2j}, \cdots, \ \mu_{cj})^T$. Based on (2), the fuzziness of the trained classifier on $\mathbf{x}_j$ is given by

$$E(\mu_j) = -\frac{1}{c}\sum_{i=1}^{c}(\mu_{ij} \ \log \ \mu_{ij} + (1-\mu_{ij}) \ \log(1-\mu_{ij})) \tag{6}$$

and furthermore, the fuzziness of the trained classifier can be given as follows.

Definition 2.2. Let the membership matrix of a classifier on the N training samples with c classes be $\mathbf{U} = (\mu_{ij})_{c\times N}$. The fuzziness of the trained classifier is given by

$$E(U) \ = -\frac{1}{cN}\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij} \ \log \ \mu_{ij} + (1-\mu_{ij}) \ \log(1-\mu_{ij})) \tag{7}$$

Equation (5) defines the fuzziness of a trained classifier which has fuzzy vector output. It plays a central role for investigating the classifier's generalization based on fuzziness. From the above definition one can view that the fuzziness of a trained classifier is actually defined as the averaged fuzziness of the classifier' outputs on all training samples. In other words, it is the training fuzziness of the classifier. The most reasonable definition of a classifier's fuzziness should be the averaged

fuzziness over the entire sample space including both training samples and unseen testing samples. However, the fuzziness for unseen samples is generally unknown, and for any supervised learning problem, there is a well-acknowledged assumption, that is, the training samples have a distribution identical to the distribution of samples in the entire space. It indicates the reasonability that we use (5) as the definition of a classifier's fuzziness.

One may argue whether the simple model (5) of averaged fuzziness for classifier outputs can exactly describe the uncertainty of classifier training since the mean is the simplest statistical feature. In fact we can choose a complex formula to model the training fuzziness. The model is dependent on the training sample distribution for computing the fuzziness. It is verified that the complex model can bring more improvement of classifier performance in comparison with the averaged model. We will report this part of research work separately.

3. Fuzziness categorization and its relation to misclassification

In this section we will experimentally observe the relationship between misclassified samples and their fuzziness for a given well-trained classifier with fuzzy vector output. Two classifiers are selected for the experimental demonstration. They are Fuzzy k-Nearest Neighbor (FKNN) [27] and Extreme Learning Machine (ELM) [28, 29]. The two classifiers and their training and testing processes are briefly reviewed as follows.

3.1. Fuzzy k-nearest neighbor (FKNN)

It belongs to a lazy learning mechanism and has not any general training process. Fuzzy k-NN method was first designed and proposed by [27] which provides a vector of class membership, where components of the vector are number in [0,1]. Fuzzy K-NN assigns memberships of class to the sample rather than a particular class as compared to traditional k-NN method. The below formula provides class memberships to the sample as the function of the sample's distances from its K-NN training samples.

$$ui(x) = \frac{\sum_{j=1}^{K} uij \left(1/||x - xj||^{2/(m-1)}\right)}{\sum_{j=1}^{K} \left(1/||x - xj||^{2/(m-1)}\right)} \tag{8}$$

Where $u_i(x)$ in (6) is the membership of the sample x to the *i-th* class i.e. $(u_1(x), u_2(x), \ldots, u_c(x))$. In this paper we used Euclidean metric to calculate the distance between test samples and its nearest training samples. Many other techniques can be use for calculating the distance. In (6) $u_i j$ is the membership value of the *j-th* neighbor belonging to the *i-th* class. The factor m used in the Equation (6) is to measure how much the distance is weighted, which indicates the contribution of each neighbor to the membership value. In the experiments we set the value of variable $m = 2$

3.2. Extreme Learning Machine (ELM)

ELM [28, 29] is a three layer feed-forward neural network in which the weights (r_{ij}) between input and hidden layers are randomly chosen and the weights (β_j) between hidden and output layers are obtained by solving a pseudo-inverse matrix. The training task is to determine the connection weights r_{ij} and $\beta_j\,(i = 1, 2, \cdots, n; j = 1, 2, \cdots, m)$. Since the weights $r_{ij}\,(i = 1, 2, \cdots, n; j = 1, 2, \cdots, m)$ are randomly selected, the training task is reduced to determine $\beta_j\,(j = 1, 2, \cdots, m)$ only. Suppose that the set of training data contains N examples which can be expressed as an input matrix A (with N rows and n columns) and a N-dimensional output vector b, respectively denoted by

$$A_{N\times m} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \ldots & a_{Nn} \end{pmatrix} \quad \text{and} \quad b_{N\times 1} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} \tag{9}$$

The weights between input layer and hidden layer are expressed as a matrix with n rows and m columns, i.e., $R = \left(r_{ij}\right)_{n\times m}$, and the weights between hidden layer and output layer are denoted as a m-dimensional vector, i.e., $\beta = (\beta_1, \beta_2, \cdots, \beta_m)^T$. Let

$$\underset{N\times m}{S} \triangleq \underset{N\times m}{A}\ \underset{n\times m}{R} = (s_{ij})_{N\times m} \text{ and}$$

$$\underset{N\times m}{H} \triangleq (f(S_{ij}))_{N\times m} = (h_{ij})_{N\times n},$$

where $f(x) = \frac{1}{1+e^{-x}}$ denotes the Sigmoid function. Then the training task of the ELM is transferred to solve the following system of linear equations $H\beta = b$, which is equivalent to the following optimization problem: $\min_{\beta\in R^m} \left\| \underset{N\times 1}{b} - \underset{N\times m}{H}\ \underset{m\times 1}{\beta} \right\|^2$. Huang

Table 1
List of data sets

Data set	Total sample	Input features	Classes
Automobile	159	15	6
Autompg	392	5	3
Cleveland	297	5	5
Wine q.	4898	11	7
Yeast	1484	8	10
Ecoli	336	5	8
Vehicle	846	8	4
Glass	214	9	6
Parkinson	195	22	2
Sonar	208	60	2

et al. [28, 29] suggested to use the minimum-norm minimum least square solution as the final one: $\min_{\|\beta\|}\left(\min_{\beta\in R^m}\left\|\underset{N\times 1}{b}-\underset{N\times m}{H}\ \underset{m\times 1}{\beta}\right\|^2\right)$. Its solution can be expressed $\beta = H^+ b$, where H^+ denotes the plus-generalized inverse this is unique for an matrix H.

Ten benchmark data sets are taken from UCI machine learning repository [30] to experimentally acquire a statistical relation between fuzziness of classifier output and the rate of misclassification. The 10 datasets and their main features are listed in Table 1.

Specifically our first experiment including 7 steps is described as follows:

i. Randomly divide the samples as training and testing sets where the training set occupies a proportion of 70% in all samples.
ii. Training a classifier based on ELM program.
iii. For each sample both in the training set and in the testing set, obtaining the fuzzy vector output based on Fuzzy k-NN and ELM respectively.
iv. Compute the fuzziness for each output.
v. Sort the samples based on the quantity of fuzziness in training set and in testing set respectively.
vi. Based on the sorting, categorizing the training set (respectively the testing set) into three parts, i.e., high-fuzziness, mid-fuzziness, and low-fuzziness subsets.
vii. Observing the correct rate of classification on each of the three parts both for training and for testing.

The flowchart of experiment-1 is shown in Fig. 2.

As an illustration, the experimental results on two datasets by using ELM and F-KNN are shown in Fig. 3 (a–h)

Our second experiment for a given dataset is designed as follows:

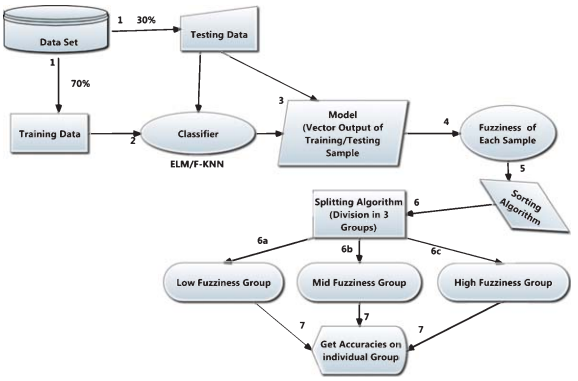

Fig. 2. Flowchart of experiment-1.

i. Randomly selecting 70% samples from the dataset as the training set.
ii. Train a classifier based on ELM (resp. Fuzzy k-NN) program.
iii. For each sample both in the training set and in the testing set, obtaining the fuzzy vector output based on Fuzzy k-NN and ELM respectively.
iv. Compute the averaged fuzziness both in training set and testing set.
v. Evaluate the correct rate of classification both in training set and in testing set.
vi. Repeating the process N times (N is a given threshold) from step 1 to 5.
vii. Sort the fuzziness in training set and in testing set respectively.
viii. Observe the changing tendency of correct rate of classification with the quantity of fuzziness both for training and for testing.

Flowchart of experiment-2 is shown in Fig. 4.

Illustration in two datasets is shown in Fig. 5(a–h) where x-axis demonstrates the No. of rounds/times.

The paired T-test demonstrates that the classification performance difference between the classifiers of Fuzzy k-NNs with higher fuzziness and with lower fuzziness is statistically significant on all the data sets. Similarly the difference for ELM classifier is also statistically significant by paired T-test.

Both experiment-1 and experiment-2 are designed for a dataset to observe the relationship between the fuzziness of a classifier and the correct classification rate of a set of samples. An analysis on results of above-mentioned two experiments is given below. From Figs. 3 and 5 one can clearly view that the correct rate of classification both in training set and in testing set is decreasingly changing with the increase of fuzziness.

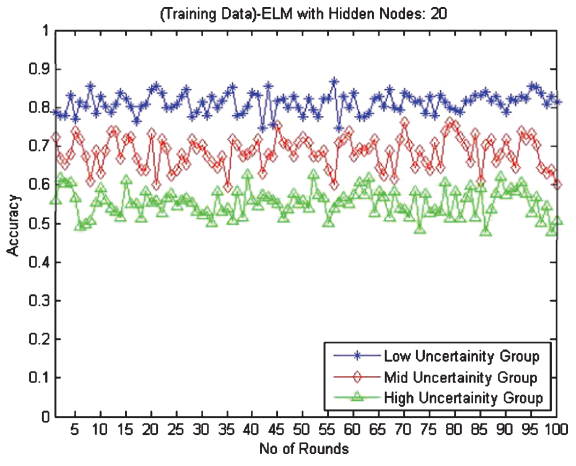

Fig. 3a. ELM-Vehicle Data Set (Training Accuracy).

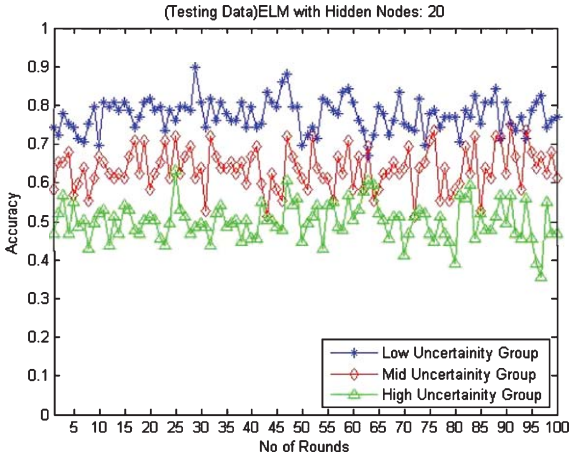

Fig. 3b. ELM-Vehicle Data Set (Testing Accuracy).

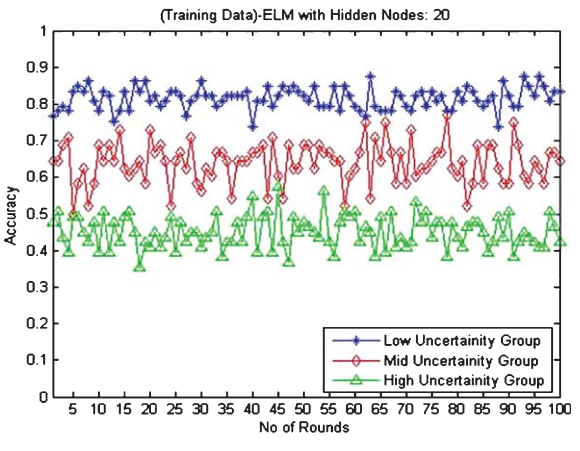

Fig. 3c. ELM-Cleveland Data Set (Training Accuracy).

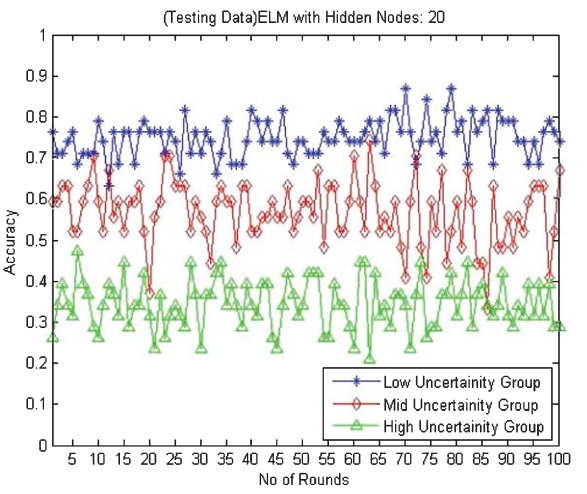

Fig. 3d. ELM-Cleveland Data Set (Testing Accuracy).

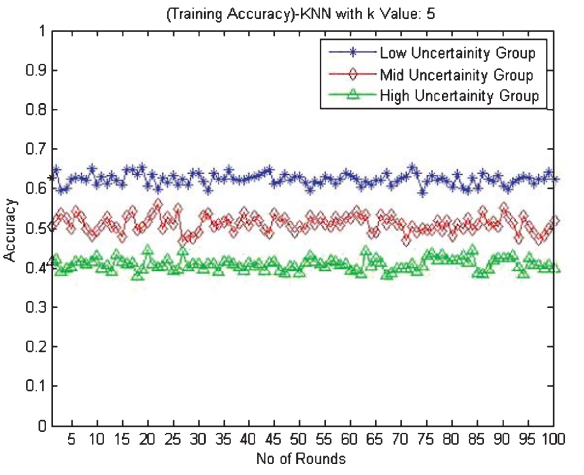

Fig. 3e. F-KNN-Wine Data Set (Training Accuracy).

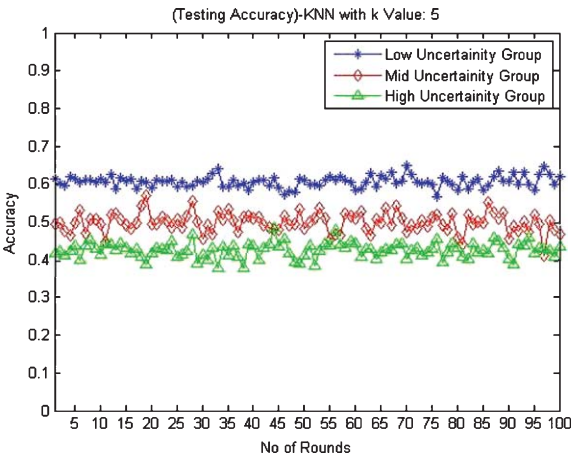

Fig. 3f. F-KNN-Wine Data Set (Testing Accuracy).

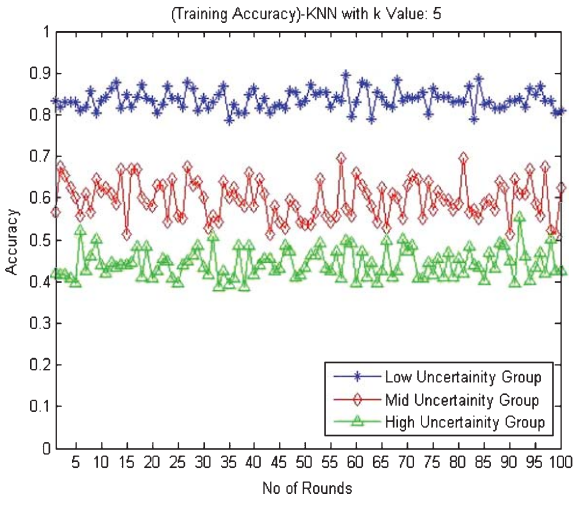

Fig. 3g. F-KNN-Vehicle Data Set (Training Accuracy).

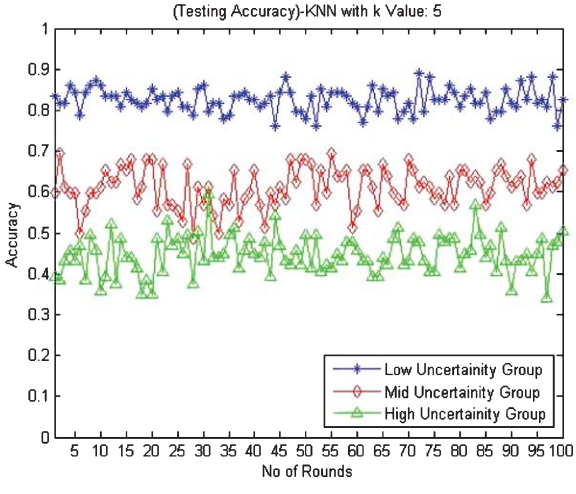

Fig. 3h. F-KNN-Vehicle Data Set (Testing Accuracy).

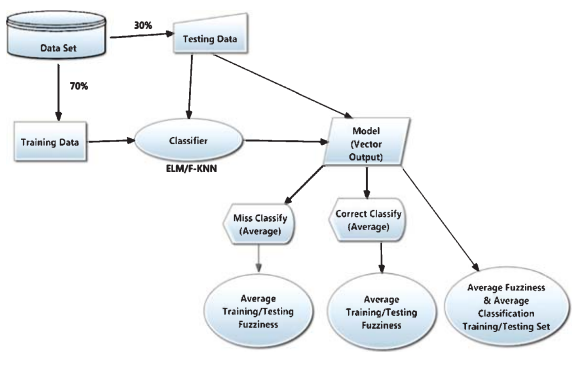

Fig. 4. Flow Chart of Experiment-2.

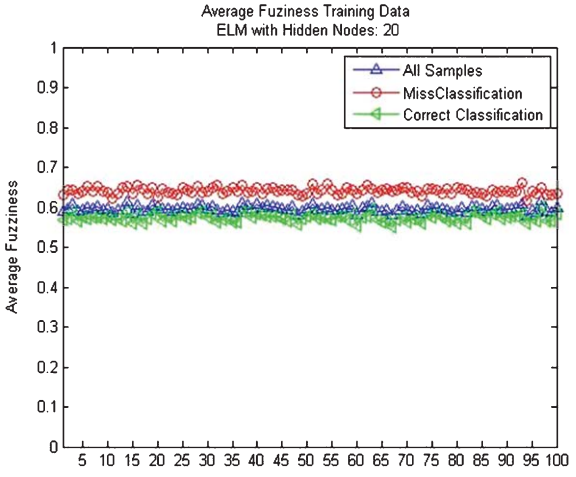

Fig. 5a. ELM-Vehicle Dataset (Training Data).

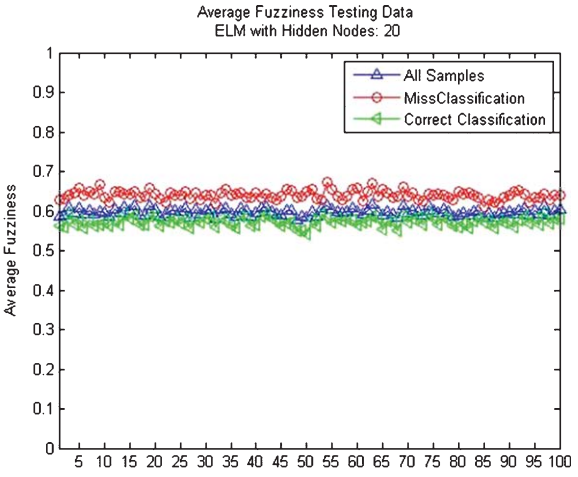

Fig. 5b. ELM-Vehicle Dataset (Testing Data).

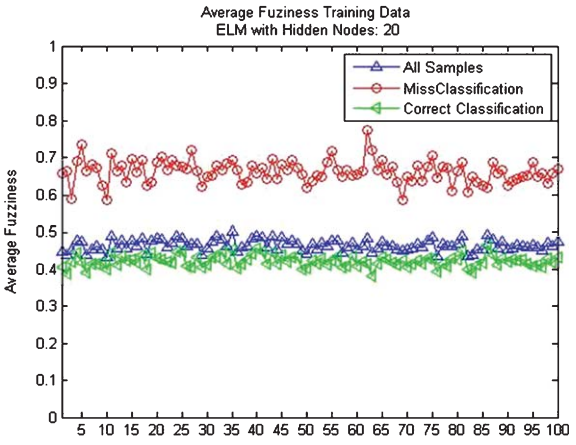

Fig. 5c. ELM-Autompg Dataset (Training Data).

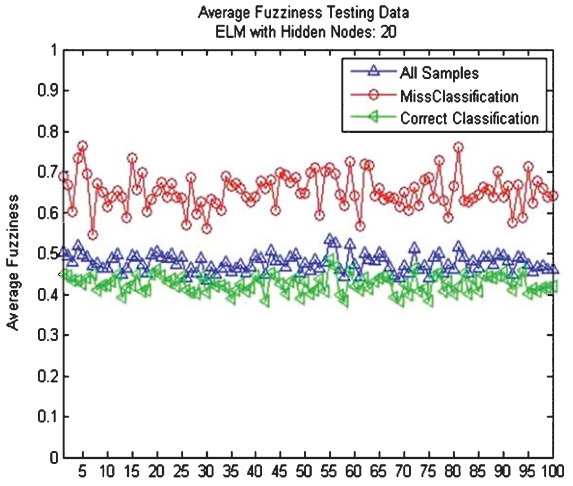

Fig. 5d. ELM-Autompg Dataset (Testing Data).

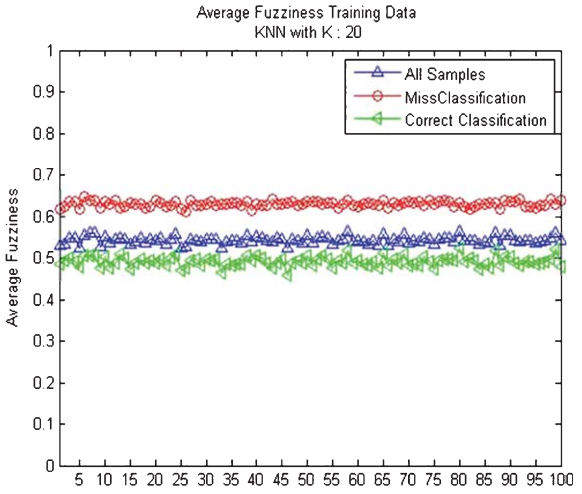

Fig. 5e. FKNN-Vehicle Dataset (Training Data).

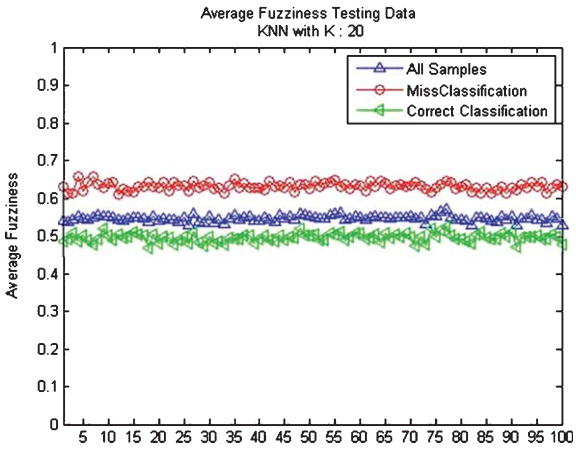

Fig. 5f. FKNN-Vehicle Dataset (Testing Data).

In other words, given a well trained classifier, a sample with high fuzziness of outputted fuzzy vector has the risk of misclassification more than a sample with low fuzziness of outputted fuzzy vector. We can explain this phenomenon from the viewpoint of boundary points. It is observed that samples with higher fuzziness are near to the classification boundary while samples with lower fuzziness are relatively far from the classification boundary. More experiments for different types of classifiers confirm this observation. In fact we have the following Proposition 1. Due to the page length limit, we omit the proof.

Proposition 1. *For a 2-class problem, let* D_1 *be the distance between the sample* $\mathbf{X}_1$ *and the classification boundary, while* D_2 *be the distance between the sample* $\mathbf{x}_1$ *and boundary. Moreover,* μ *and* σ *are the outputs of the classifier on* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*, respectively. If* $D_1 \leq D_2$*, then the fuzziness of* $\mathbf{x}_1$ *is no less than that of* $\mathbf{x}_2$*, i.e.,* $E(\mu) \geq E(\sigma)$ *where E is the fuzziness formula defined in section 2.*

Boundary samples are considered to have a key impact on classifier performance [31, 32]. This is essentially consistent with the idea of AdaBoost [33] which assigns heavier weights to the training samples that are hard to train.

In summary, a sample with high fuzziness is possibly a boundary point more than a sample with low fuzziness is. Generally speaking, for a well trained classifier, samples near to boundary have the risk of misclassification more than samples far from boundary have.

4. A divide-and-conquer strategy based on fuzziness categorization

Section 3 indicates that the risk of misclassification becomes higher as the fuzziness of training samples gets larger while the risk is relatively decreasing as the fuzziness of training samples gets statistically smaller. This analysis on misclassification risk inspires us to separate samples with high fuzziness from samples with non-high fuzziness. The experiments in the previous section have confirmed that samples with high fuzziness have a correct rate of classification less than samples with low fuzziness have. That is, samples with higher fuzziness are more difficult to be correctly classified in comparison with samples having lower fuzziness. From the viewpoint of boundary points discussed in previous section, the explanation can be that boundary points are more difficult to be correctly classified in comparison

with inner points. It is worth noting that for most classification problems boundary points have an essential impact on classification performance more than inner points have. In this way, for improving the classification performance, we propose a new idea which basically is a strategy of Divide-and-Conquer. That is, we will cope with the samples with low fuzziness usually while cope with samples with high fuzziness specially.

According to the magnitude of fuzziness, all samples are categorized as three groups, as discussed in the experiments of Section 3. One group is of high fuzziness while the other two groups are of mid and low fuzziness respectively. Our strategy is adding the testing samples with low fuzziness and the predicted labels of these samples in the original training set, which generates a new training set. Then using the new training set, we re-train a classifier to predict the testing samples (with high fuzziness). It is expected that our proposed strategy can result in an improvement of classification performance on the category of samples with high fuzziness.

Specifically our algorithm based on the Divide-and-Conquer strategy is listed as follows.

i. Randomly Split the entire dataset into two subsets: training and testing.
ii. Training a classifier based on ELM/Fuzzy k-NN program.
iii. For each sample both in the training set and in the testing set, obtaining the fuzzy vector output given by Fuzzy k-NN and ELM respectively.
iv. Computing the fuzziness for each output, and writing down the training accuracy and testing accuracy.
v. Sorting the samples based on the quantity of fuzziness in training set and in testing set respectively.
vi. Based on the sorting, categorizing the training set (respectively the testing set) into three parts, i.e., high-fuzziness, mid-fuzziness, and low-fuzziness subsets.
vii. Evaluating the correct rate of classification on each of the three parts both for training and for testing.
viii. Adding the testing samples with low fuzziness in the original training set.
ix. Retraining an ELM/Fuzzy k-NN on the new training set.
x. Using the ELM and Fuzzy k-NN to predict samples of testing set, and writing down the training accuracy and testing accuracy.
xi. Making a comparison between the accuracies obtained in step-(iv) and step-(x) respectively.

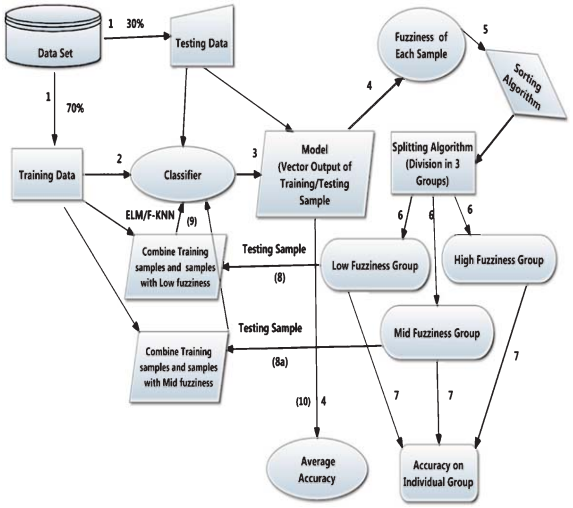

Fig. 6. Flowchart of Experiment-3 (Divide & Conquer).

Table 2a
ELM, #. of Hidden Nodes: 20 Uncertainty Type: Fuzziness

Data Set	$\text{Train}_{\text{Acc}}$, Test_{Acc}	(Adding the Low fuzziness group of samples which are testing samples and their predicted labels in the original training set)
		$\text{Train}_{\text{Acc}}$, Test_{Acc}
Autompg	(80.7692, 79.8319)	(82.3344, 81.9328)
Cleveland	(65.0485, 50.5495)	(65.8333, 52.7473)
Wine Q.	(53.9988, 52.9212)	(54.55, 53.8723)
Yeast	(59.0338, 59.9109)	(60.6816, 63.4744)
Ecoli	(91.1638, 79.3269)	(91.8819, 81.7308)
Vehicle	(70.0847, 63.0859)	(72.0845, 66.0156)
Automobile	(69.5455, 52.0408)	(69.9219, 58.1633)
Parkinson	(89.6296, 85.5)	(90.3136, 91.6667)
Glass	(73.6486, 64.2424)	(75.4335, 67.7273)
Sonar	(80.5556 60.9375)	(73.9130 70.3125

The algorithm flowchart is shown Fig. 6

The 10 datasets selected from UCI machine Learning Repository are used again to verify the performance of our proposed algorithm. The experimental results are listed in Table 2a and b.

From Table 2a and b, one can clearly see an improvement of testing accuracy for every dataset, which confirms the effectiveness of our proposed approach. Additional experiments on a lot of datasets shows the same result of improvement which really benefits from the separation of high fuzziness samples from low fuzziness samples and the corresponding Divide-and-Conquer strategy. One may argue that the improvement shown in Table 2a and b is not very significant and a more effective Divide-and-Conquer strategy may be

Table 2b
Fuzzy k-NN (#. of Neighbors: 5), $m = 2$, Uncertainty Type: Fuzziness

Data Set	$\text{Train}_{\text{Acc}}$, Test_{Acc}	(Adding the Low fuzziness group of samples which are testing samples and their predicted labels in the original training set)
		$\text{Train}_{\text{Acc}}$, Test_{Acc}
Autompg	(72.5275,75.7595)	(78.9916, 78.9916)
Cleveland	(47.0874, 51.0162)	(54.9451,53.8462)
Wine Q.	(50.3503, 50.7186)	(50.5433, 51.2908)
Yeast	(51.5942, 51.6323)	(51.5942, 52.5612)
Ecoli	(81.4655, 84.0020)	(84.0020, 86.0577)
Vehicle	(60.6780, 60.0265)	(60.6780,60.0365)
Automobile	(70.0000, 67.6531)	(71.4286, 71.4286)
Parkinson	(93.3333, 95.0000)	(98.3333, 98.3333)
Glass	(65.5405, 68.3763)	(72.7273, 72.7273)
Sonar	(79.8611 79.6875)	(73.3333 82.8125)

necessary. A new Divide-and-Conquer strategy (based on the separation of high fuzziness samples from low fuzziness samples) possibly results in a bigger improvement. In fact the improvement of accuracy resulted from the Divide-and-Conquer strategy is focusing on the samples with high fuzziness but these samples are inherently difficult to be correctly classified due to the side effect of classifiers. We do not expect an accuracy improvement which is coming from handling inner samples.

The essential part of this algorithm is to add the testing samples with low fuzziness and their predicted labels in the original training set, which enlarges the training set but possibly includes some error samples (since the testing accuracy on low fuzziness group is not 100%). To make clear which part (low or mid fuzziness group added in the original training set) plays a more important role with respect to the classification performance improvement, we update the above developed Divide-and-Conquer algorithm by revising step (viii) as follows.

(viii-a). By adding the testing samples having low and mid fuzziness and their predicted labels in the original training set.

(viii-b). By adding the testing samples having the mid fuzziness and their predicted labels in the original training set.

The experimental results corresponding to revision (viii-a) and (viii-b) by ELM classifier is shown in Table 3.

Table 3 indicates that, regarding adding low and mid fuzziness group to the original training set, the low fuzziness group plays a more essential role than the mid fuzziness group for classification performance improvement. Table 3 also indicates that the improvement through only adding mid fuzziness group to the original training set is very little.

We finally give some remarks on the fuzziness categorization based Divide-and-Conquer strategy and its algorithm. The essential idea of the whole paper is dependent on the difference between low and high fuzziness groups of samples

1) The difference implicitly tells us some meaningful ways to improve the classification performance.
2) Samples users are really interested in those with high fuzziness. The difference is to make users pay particular attention to samples with high fuzziness and to tell users that the classification for samples with low fuzziness is much possibly correct even they use a simple trained classifier.
3) Due to the classification performance on the low fuzziness group of samples, it is usual to add them in the original training set and then to improve the

Table 3
ELM, #. of Hidden Nodes: 20 Uncertainty Type: Fuzziness

Dataset	$\text{Train}_{\text{Acc}}$, Test_{Acc}	By Adding Low Fuzziness Group	By Adding the Mid Fuzziness Group	By Adding the Low & Mid Fuzziness Group
		$\text{Train}_{\text{Acc}}$, Test_{Acc}	$\text{Train}_{\text{Acc}}$, Test_{Acc}	$\text{Train}_{\text{Acc}}$, Test_{Acc}
Autompg	82.0513, 75.6303	83.9117, 80.6723	83.9117, 76.4706	84.7701, 78.9916
Cleveland	64.0777, 52.7473	70.0000, 54.9451	68.7500, 51.6484	71.1027, 53.8462
Wine Q.	53.8821, 53.0571	58.9744, 54.1440	57.9940, 53.4647	61.5509, 53.6005
Yeast	59.1304, 56.7929	65.5860, 58.3519	64.7548, 57.6837	67.5532, 57.0156
Ecoli	88.3621, 83.6538	90.7749, 86.5385	89.6679, 84.6154	91.9192, 85.5769
Vehicle	66.6102, 62.5000	70.8455, 65.6250	68.9504, 61.7188	72.4000, 63.2813
Automobile	72.7273, 44.8980	75.0000, 48.9796	72.6563, 44.8980	69.5035, 46.9388
Parkinsons	91.0370, 85.1667	93.5386, 87.0000	92.8326, 85.3333	92.8874, 86.3333
Glass	72.9730, 69.6970	76.3006, 74.2424	76.8786, 66.6667	80.4233, 72.7273
Sonar	80.5556, 60.9375	73.9130, 70.3125	77.0186, 62.5000	72.8261, 68.7500

performance in the group of samples with high fuzziness.

4) Basically the proposed technique of adding low fuzziness samples and their predicted labels in the original training set belongs to the field of semi-supervised learning where some samples with unknown labels participate in the training process.
5) It is interesting to study other methodologies and algorithms of specially handling samples with high fuzziness. Possibly ensemble learning and evolutionary computing are promising and effective ways to particularly handle high fuzziness samples although they have a larger computational complexity.
6) Incorporating fuzzy techniques into learning processes for model performance improvement has been conducted over several decades. One can find many approaches from the existing references. For example, a multiple fuzzy NN classifier system based on mutual information and fuzzy integral [34], FPGP technique with parameters based on fuzzy coefficient [35], multilevel thresholding method based on maximum fuzzy entropy for extracting edge information [36] and classifying E-Health Projects based on advanced cluster techniques and fuzzy theories [37] have been proposed. A scheme to design adaptive output-feedback controller for uncertain nonlinear systems, where fuzzy logic systems are used to approximate the nonlinear function is proposed in [38]. In [39] author proposed the image classification method which is based on positive and negative fuzzy rule system using extreme learning machine. However, for the first time this paper establishes the linkage between the outputted fuzziness and the generalization of a classifier.

This paper selects the ELM and fuzzy k-NN as an illustration of classifier. In fact algorithms developed in this paper are suitable for any type of classifier with fuzzy vector output. There is no obvious evidence to show that the classifier type is sensitive to the Divide-and-Conquer algorithm.

The main contributions of this paper include finding the relationship between classifier's fuzziness and its misclassification rate; verifying that samples with higher fuzziness exhibit higher risk of misclassification; and developing a divide-and-conquer learning algorithm for classifier performance improvement based on the fuzziness categorization.

5. Conclusions

It is confirmed, for a well-trained classifier which outputs a membership vector, samples with higher fuzziness outputted by the classifier mean a bigger risk of misclassification. Separately handling the samples with low and high fuzziness (i.e. specially handling the high fuzziness sample category) is an effective way to promote the correct classification rate. This paper proposes a Divide-and-Conquer strategy by adding the low fuzziness samples in the original training set and then forming a new training set. It is a semi-supervised learning approach of which the reasonability is that the quality enlargement of training set and retraining is helpful to classification performance improvement. Experimental results show that the approach is effective.

Acknowledgments

This research is supported by National Natural Science Fund of China (61170040 and 71371063) and Basic Research Project of Knowledge Innovation Program in Shenzhen (20150307181003).

References

[1] X.D. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. Mclachlan, A. Ng, B. Liu, P.S. Yu, Z.H. Zhou, M. Steinbach, D.J. Hand and D. Steinberg, Top 10 algorithms in data mining, *Knowledge Information Systems* **14** (2008), 1–37.

[2] Y. Su, G. Agrawal, J. Woodring, K. Myers, J. Wendelberger and J. Ahrens, Effective and efficient data sampling using bitmap indices, *Cluster Computing* (2014), 1–20.

[3] G.U. Baohua, F. Hu, H.U. Feifang and L.I.U. Huan, Sampling and its application in data mining: A survey, *National University of Singapore, School of Computing* **31** (2000), 1–26.

[4] M. Stone, Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **36**(2) (1974), 111–147.

[5] O. Chapelle, V. Vapnik, O. Bousquet and S. Mukherjee, Choosing multiple parameters for support vector machines, *Machine Learning* **46**(1–3) (2002), 131–159.

[6] W.Y. Ng, P.F. Chan, D.S. Yeung and C.C. Tsang, Quantitative study on the generalization error of multiple classifier systems, *in Proc Inter Conf on SMC*, 2005, pp. 889–894.

[7] D.S. Yeung, W.Y. Ng, D.F. Wang, C.C. Tsang and X.Z. Wang, Localized generalization error model and its application to architecture selection for radial basis function neural network, *IEEE Trans on Neural Networks* **18**(5) (2007), 1294–1305.

[8] P.K. Chan, D.S. Yeung, W.Y. Ng, C.M. Lin and N.K. Liu, Dynamic fusion method using localized generalization error model, *Information Science* **217** (2012), 1–20.

[9] A. Agarwal and J.C. Duchi, The generalization ability of online algorithms for dependent data, *IEEE Trans on Information Theory* **59**(1) (2013), 573–587.
[10] C.C. Gavin and L.C. Nicola, On over-fitting in model selection and subsequent selection bias in performance evaluation, *Journal of Machine Learning Research* **11** (2010), 2079–2107.
[11] V.N. Vapnik, *Estimation of dependences based on empirical data*, Springer Verlag, New York, 1982.
[12] B.E. Boser, I.M. Guyon and V.N. Vapnik, A training algorithm for optimal margin classifiers, *in Proc 5th Annu Workshop Comput Learn Theory*, 1992, pp. 144–152.
[13] N. Littlestone and M.K. Warmuth, The weighted majority algorithm, *Information Computation* **108**(2) (1994), 212–261.
[14] R.E. Schapire, Y. Freund, P. Bartlett and W.S. Lee, Boosting the margin: A new explanation for the effectiveness of voting methods, *Annals of Statistics* **26**(5) (1998), 1651–1686.
[15] S. Decherchi, S. Ridella, R. Zunino, P. Gastaldo and D. Anguita, Using unsupervised analysis to constrain generalization bounds for support vector classifiers, *IEEE Transactions on Neural Networks* **21**(3) (2010), 424–438.
[16] O. Ludwig, U. Nunes, B. Ribeiro and C. Premebida, Improving the generalization capacity of cascade classifiers, *IEEE Transactions on Cybernetics* **43**(6) (2013), 2135–2146.
[17] X.Z. Wang and C.R. Dong, Improving generalization of fuzzy IF-THEN rules by maximizing fuzzy entropy, *IEEE Transactions on Fuzzy System* **17**(3) (2009), 556–567.
[18] X.Z. Wang, L.C. Dong and J.H. Yan, Maximum ambiguity-based sample selection in fuzzy decision tree induction, *IEEE Transactions on Knowledge and Data Engineering* **24**(8) (2012), 1491–1505.
[19] Y. Yuan and M.J. Shaw, Induction of fuzzy decision trees, *Fuzzy Sets and Systems* **69** (1995), 125–139.
[20] D. Dubois and H. Prade, Rough fuzzy sets and fuzzy rough sets, *International Journal of General Systems* **17**(2-3) (1990), 191–209.
[21] L.A. Zadeh, Probability measures of fuzzy events, *Journal of Mathematical Analysis and Applications* **23** (1968), 421–427.
[22] A. De Luca and S. Termini, A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory, *Information and Control* **20** (1972), 301–312.
[23] A. De Luca and S. Termini, Entropy of L-fuzzy sets, *Information and Control* **24** (1974), 55–73.
[24] G. Klir, where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like? *Fuzzy Sets and Systems* **24**(2), (1987), 141–160.
[25] G. Klir and T. Folger, *Fuzzy sets, uncertainty and information*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[26] D.S. Yeung and E. Tsang, Measures of fuzziness under different uses of fuzzy sets, *Advances in Computational Intelligence Communications in Computer and Information Science*, vol. 298, 2012, pp. 25–34.
[27] J.M. Keller, M.R. Gray and J.A. Givens, A fuzzy K-nearest neighbor algorithm, *IEEE Transactions on Systems, Man and Cybernetics* **15**(4) (1985), 580–585.
[28] G.B. Huang, L. Chen and C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Transactions on Neural Networks* **17**(4) (2006), 879–892.
[29] G.B. Huang, D.H. Wang and Y. Lan, Extreme learning machines: a survey, *International Journal of Machine Learning and Cybernetics* **2**(2) (2011), 107–122.
[30] K. Bache and M. Lichma, *UCI Macine Learning Repository*, Online Available at http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science 2013.
[31] C. Lee and D.A. Landgrebe, Decision boundary feature extraction for neural networks, *IEEE Transactions on Neural Networks* **8**(1) (1997), 75–83.
[32] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun and V. Vapnik, Boosting and other ensemble methods, *Neural Computing* **6**(6) (1994), 1289–1301
[33] O.D. Richard, P.E. Hart and D.G. Stork, *Pattern Classification*, John Wiley & Sons, 2012.
[34] L. Wang, An improved multiple fuzzy NNC system based on mutual information and fuzzy integral, *International Journal of Machine Learning and Cybernetics* **2**(1) (2011), 25–36.
[35] G.S. Mahapatra, T.K. Mandal, and G.P. Samanta, A production inventory model with fuzzy coefficients using parametric geometric programming approach, *International Journal of Machine Learning and Cybernetics* **2**(2) (2011), 99–105.
[36] P.P. Guan and H. Yan, A hierarchical multilevel thresholding method for edge information extraction using fuzzy entropy, *International Journal of Machine Learning and Cybernetics* **3**(4) (2012), 297–305.
[37] P. D'Urso, L. De Giovanni and P. Spagnoletti, A fuzzy taxonomy for e-Health projects, *International Journal of Machine Learning and Cybernetics* **4**(5) (2013), 487–504.
[38] Y. Li, T. Li and S. Tong, Adaptive fuzzy modular backstepping output feedback control of uncertain nonlinear systems in the presence of input saturation, *International Journal of Machine Learning and Cybernetics* (2013).
[39] W. Jun, W. Shitong and F.-l. Chung, Positive and negative fuzzy rule system, extreme learning machine and image classification, *International Journal of Machine Learning and Cybernetics* **2**(4) (2011), 261–271.