Discovering the Relationship Between Generalization and Uncertainty by Incorporating Complexity of Classification

Xi-Zhao Wang, *Fellow, IEEE*, Ran Wang, *Member, IEEE*, and Chen Xu

***Abstract*—The generalization ability of a classifier learned from a training set is usually dependent on the classifier's uncertainty, which is often described by the fuzziness of the classifier's outputs on the training set. Since the exact dependency relation between generalization and uncertainty of a classifier is quite complicated, it is difficult to clearly or explicitly express this relation in general. This paper shows a specific study on this relation from the viewpoint of complexity of classification by choosing extreme learning machines as the classification algorithms. It concludes that the generalization ability of a classifier is statistically becoming better with the increase of uncertainty when the complexity of the classification problem is relatively high, and the generalization ability is statistically becoming worse with the increase of uncertainty when the complexity is relatively low. This paper tries to provide some useful guidelines for improving the generalization ability of classifiers by adjusting uncertainty based on the problem complexity.**

***Index Terms*—Complexity of classification, extreme learning machine, generalization, uncertainty.**

I. Introduction

CLASSIFICATION problem, as the central part in the fields of pattern recognition and data mining, refers to a task of assigning objects to one of several predefined class labels. Given a set of objects, the mathematical model of classification problem is a discrete-valued function that maps each object to a class label. Usually, the process of determining the discrete-valued function from a training set is called learning while the process of using the determined function to classify a new object is called reasoning [1]–[5].

For a classification problem with c classes, the reasoning result is generally a c-dimensional vector. According to the output forms of the reasoning process, existing learning algorithms can be classified into two categories. In one category, the c-dimensional output vector contains one component of value 1 and other components of value 0. In this situation, the class label corresponding to the component 1 will be the reasoning result. This kind of algorithms are known as crisp-output algorithms, such as traditional support vector machine (SVM) [6]–[10], decision tree (DT) [11], [12], etc. In the other category, the c-dimensional output vector contains components of real values within the interval [0, 1]. In this situation, the class label corresponding to the maximum component will be the reasoning result. If the maximum is attained at more than one component, a special strategy will be designed to determine the final result. This kind of algorithms are acknowledged as uncertain-output algorithms, such as k-nearest neighbor [2], Bayesian probability model [2], back-propagation (BP) methods for training feed-forward neural networks [13]–[16], etc.

Obviously, crisp-output algorithms are special cases of uncertain-output algorithms. If an algorithm belongs to the crisp category, then it belongs to the uncertain category, however, it is not true conversely. Most crisp-output algorithms can be extended to uncertain-output algorithms, such as fuzzy SVM [17], fuzzy DT [18], etc. In this paper, we will intensively investigate the uncertain-output algorithms, which highlight the argument that uncertainty does exist in the learning and reasoning processes.

On the other hand, generalization of a classifier is defined as the rate of the correctly classified objects that are not in the training set. It is the most important index for evaluating a classification algorithm since the ultimate goal for developing a classification model is to achieve high prediction accuracy on unseen cases. Usually, the generalization of a classifier depends on multiple factors.

1) The mathematical model, which has a direct impact on both the training accuracy and testing accuracy.
2) The algorithm for training the model parameters, which is sensitive to the prediction results.

Manuscript received July 4, 2016; revised November 28, 2016; accepted January 11, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61402460, Grant 61472257, Grant 61170040, and Grant 71371063, in part by the Basic Research Project of Knowledge Innovation Program in Shenzhen under Grant JCYJ20150324140036825, in part by the Guangdong Provincial Science and Technology Plan Project under Grant 2013B040403005, and in part by the HD Video Research and Development Platform for Intelligent Analysis and Processing in Guangdong Engineering Technology Research Centre of Colleges and Universities under Grant GCZX-A1409. *(Corresponding author: Ran Wang.)*

X.-Z. Wang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: xizhaowang@ieee.org).

R. Wang and C. Xu are with the College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, China (e-mail: wangran@szu.edu.cn; xuchen@szu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2653223

3) *The data distribution:* In supervised learning, there is a fundamental assumption that the training data has the same distribution as the testing data. The learning scheme that does not follow this fundamental assumption is referred to as transfer learning [19], which is out of the scope of this paper.

Many research efforts have been made to improve the generalization of a classifier by considering different factors. In this paper, we consider a particular model parameter, i.e., the uncertainty of the classifier's outputs, which has been proven in [20] to have a close relationship with the generalization of classifier. It has been shown in [20] that the uncertainty of the classifier's outputs has a close relationship with the generalization capability. However, this relationship is difficult to express explicitly for general cases. In order to further investigate this relationship, in this paper, we take into account a new index, i.e., complexity of classification, which can be measured in different ways [21]. To the best of our knowledge, this paper makes a first attempt to investigate the relationship between generalization and uncertainty of a classifier by incorporating the complexity of classification.

In addition, choosing an appropriate classification algorithm is also an important issue to conduct this research. It is noteworthy that any uncertain-output algorithm can be used to study the relationship between generalization and uncertainty. As the commonly used classification model for various practical problems, feed-forward neural networks will be adopted. The most notable algorithm to train a feed-forward neural network is BP. Although it has been proved in [15] and [16] that BP network has the ability to approximate any continuous function with arbitrary precision, it is often criticized to have the problems of slow convergence speed and local minima. In order to overcome these deficiencies, extreme leaning machine (ELM) has been proposed as a new training algorithm for single-hidden layer feed-forward neural network (SLFN) [22]. Differentiating from BP that iteratively tunes the weight parameters by gradient descent technique, ELM randomly chooses the weight parameters between input and hidden layers and analytically solves the weight parameters between hidden and output layers through Moore–Penrose generalized inverse [44]–[48]. Due to the extremely fast training speed and good prediction performance, ELM has been investigated intensively and extensively in the machine learning and data mining communities [23]–[26]. Based on the aforementioned advantages, we will adopt ELM as the classification algorithm in this paper. The major theoretical issues of ELM can be found in [27] and [28], and the applications of ELM to different areas, such as sparse representation can be found in [29] and [30].

The rest of this paper is organized as follows. Section II reviews ELMs. Section III introduces the dependency relation between generalization and uncertainty of classifiers. Section IV discusses the complexity of classification problems. Section V analyzes the relationship between generalization and uncertainty by incorporating a complexity index. Experiments are conducted in Section VI. Finally, conclusions are given in Section VII.

II. Extreme Learning Machine

This section will introduce ELM, which is a noniterative training algorithm for SLFNs.

A. Training of ELM

A standard SLFN for classification is a discrete function mapping samples to class labels. Given a training set that contains N arbitrarily distinct samples $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \subset \mathcal{R}^n \times \{0, 1\}^c$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ is the ith training sample, $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{ic}]$ is the label vector of $\mathbf{x}_i$, n is the number of features, and c is the number of classes. An SLFN with $\widetilde{N}$ hidden nodes and activation function $g(\mathbf{x})$ can be expressed as

$$\sum_{j=1}^{\widetilde{N}} \beta_j g\left(\mathbf{w}_j \cdot \mathbf{x}_i + b_j\right) = \mathbf{t}_i, \ \ i = 1, 2, \ldots, N \tag{1}$$

where $\mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jn}]$ is the weight linking the input nodes to the jth hidden node, b_j is the bias of the jth hidden node, β_j is the weight linking the jth hidden node to the output nodes, and sigmoid function $g(x) = (1/[1 + \exp(-x)])$ is selected as the activation function.

In ELMs, the input weights $\mathbf{w}_j$ and biases b_j are randomly chosen, and the learning can be formulated as a minimum optimization problem with a regularized term

$$\min_\beta \left\{ ||\mathbf{T} - \mathbf{H}\beta||_2^2 + \mu ||\beta||_2^2 \right\}, \ \ \mu > 0 \tag{2}$$

where $\mathbf{H}$ is the hidden layer output matrix denoted as

$$\begin{aligned} &\mathbf{H}(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{\widetilde{N}}, b_1, b_2, \ldots, b_{\widetilde{N}}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \\ &= \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g\left(\mathbf{w}_{\widetilde{N}} \cdot \mathbf{x}_1 + b_1\right) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g\left(\mathbf{w}_{\widetilde{N}} \cdot \mathbf{x}_N + b_{\widetilde{N}}\right) \end{bmatrix}_{N \times \widetilde{N}} \end{aligned} \tag{3}$$

and $\mathbf{T}$ is the label matrix denoted as

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{bmatrix}_{N \times c}. \tag{4}$$

The optimal estimation of output weights β^* can be formulated as a regularized least square problem

$$\beta^*_{\widetilde{N} \times c} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \mu\mathbf{I}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T} \tag{5}$$

where $\mathbf{I}$ is the identity matrix of suitable dimension and μ is the regularizing factor.

To this end, all the parameters $\{\mathbf{w}, b, \beta\}$ in ELM have been fixed, and the training process is finished.

ELMs have been proved to have the universal approximation capabilities [31] although the training process does not include any iteration. Under the assumption of smoothness of the underlying function, the universal approximation capability of ELMs can be guaranteed by providing a sufficiently large number of hidden nodes with certain range of $\mathbf{w}$ and b.

In comparison with BP algorithm, ELMs have a much faster training speed due to the noniterative mechanism. References show that ELMs can finish the training process thousands of times faster than BP in some scenarios, at the same time, an

acceptable learning accuracy is kept. The advantages and disadvantages of ELMs are listed in Appendix A. Furthermore, one can find many improved versions for ELMs. The computation of weights between hidden and output layers can be improved through an optimization algorithm given by Deng *et al.* [32] in order to avoid over-fitting. Rong *et al.* [33] offered a pruned ELM in which the corresponding nodes can be removed according to the information gain to reduce the correlation among classes in a large network structure. Feng *et al.* [34] proposed an EM-ELM in which the weights are not updated when a node is added, and the algorithm can update the weights and adjust the network at the same time. Furthermore, it is found that ELMs can online deal with sequential data successfully [35].

B. Generalized Inverse and Normal Equations

In ELMs, the weights between hidden and output layers are calculated by the generalized inverse [36]. We briefly review some connections between the generalized inverse and the normal equations. Originally, the training of ELMs contains two parts. The first is to randomly assign values in a specified interval to the weights between the input and hidden layers while the second is to determine the weights between the hidden and output layers by computing the generalized inverse of the matrix $\mathbf{H}$ as $\beta^* = \mathbf{H}^\dagger\mathbf{T}$. It is the minimum norm and minimum least square solution of the system of linear matrix equations $\mathbf{H}\beta = \mathbf{T}$. It is easy to prove that, if the matrix $\mathbf{H}$ is of full-rank, the solution of normal equation $\mathbf{H}^{\mathrm{T}}\mathbf{H}\beta = \mathbf{H}^{\mathrm{T}}\mathbf{T}$ is identical to $\beta^* = \mathbf{H}^\dagger\mathbf{T}$.

Noting that in Section II-A, the training process of ELMs is written as $\beta^* = (\mathbf{H}^{\mathrm{T}}\mathbf{H}+\mu\mathbf{I})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T}$, where μ is a regularizing factor. This formula is identical to $\beta^* = \mathbf{H}^\dagger\mathbf{T}$ if the regularizing factor takes value zero. It is proven in [24] that the matrix $\mathbf{H}$ is of full-rank with probability 1, and therefore, we can say that the solution of normal equation $\mathbf{H}^{\mathrm{T}}\mathbf{H}\beta = \mathbf{H}^{\mathrm{T}}\mathbf{T}$ is available with probability 1. In fact, the regularizing factor, which makes the solved weights as small as possible, has the effect to become the matrix $\mathbf{H}$ full of rank.

Practically the number of rows is much larger than the number of columns for an input data matrix. It implies that the transformation from computing $\beta^* = \mathbf{H}^\dagger\mathbf{T}$ to solving the normal system of linear matrix equations $\mathbf{H}^{\mathrm{T}}\mathbf{H}\beta = \mathbf{H}^{\mathrm{T}}\mathbf{T}$ can save much computational load, since the order of $\mathbf{H}$ is $N \times \widetilde{N}$ but the order of $\mathbf{H}^{\mathrm{T}}\mathbf{T}$ is $\widetilde{N} \times c$, where N is the number of input samples, $\widetilde{N}$ is the number of hidden layer nodes, and c is the number of classes. A lot of numerical experiments have confirmed this saving of computational load.

III. Dependency Relation Between Generalization and Uncertainty of Classifiers

In this section, we will introduce the generalization and uncertainty of a classifier. The dependency relation between generalization and uncertainty is then discussed.

A. Generalization and Uncertainty

Generally speaking, the purpose of learning is to acquire the knowledge hidden in the data. Knowledge representation, which has been well acknowledged as a bottle-neck problem in machine learning and artificial intelligence for many years, does not have a general definition but has many specific forms. A mathematical model, such as a set of IF-THEN rules or a neural network learned from a training set, can be regarded as a typical form of knowledge representation. The ability or performance of the learned model to predict unseen cases (which are not within the training set) is called generalization.

Let $\mathcal{S}$ be a finite space of samples, $F(\mathbf{x})$ be a discrete-valued function defined on $\mathcal{S}$, and $\mathbb{X}$ be a subset of $\mathcal{S}$. Based on values of $F(\mathbf{x})$ in $\mathbb{X}$, an estimator function $f(\mathbf{x})$ defined on $\mathcal{S}$ is given by using a training algorithm. The discrete-valued function $f(\mathbf{x})$ has the same value range as $F(\mathbf{x})$. Usually we call $f(\mathbf{x})$ as a classifier trained by the algorithm on $\mathbb{X}$.

Definition 1: The generalization of classifier $f(\mathbf{x})$ is defined as

$$G(f) = \frac{|\{\mathbf{x} : \mathbf{x} \in \mathcal{S} - \mathbb{X}, F(\mathbf{x}) = f(\mathbf{x})\}|}{|\mathcal{S} - \mathbb{X}|} \tag{6}$$

where $|\ |$ denotes the number of elements in a set.

Generalization is the most important index of evaluating a learned model. From mathematical viewpoint, the task of learning is to find a function $f(\mathbf{x})$ through a training set $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N \subset \mathcal{R}^n \times \{0,1\}^c$ such that $f(\mathbf{x})$ can well approximate the objective function $F(\mathbf{x})$ both at training cases and unseen cases. The difference between $F(\mathbf{x})$ and $f(\mathbf{x})$ is called generalization error, which can be measured from different angles. One method is to estimate an upper bound for it, the other is to compute $R = \int_{\mathcal{S}} [F(\mathbf{x}) - f(\mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x}$, where $p(\mathbf{x})$ is the probability density function of input $\mathbf{x}$. Experimentally, the generalization can be measured by the prediction accuracy of the classifier on a testing set.

Multiple factors have critical impacts on the generalization of a classifier.

1) *Model Selection:* It is hard to select the most appropriate model for a given classification task. When the training data is fixed, the generalizations of two models might be quite different. This is due to the data distribution, i.e., a model suitable for one type of data may not be appropriate for another type of data.
2) *Training Algorithm:* When a model is fixed, the subsequent work is to train the model parameters based on a training set. A model with a set of trained parameters has the generalization quite different from the model with another set of trained parameters.
3) *Representatives of Training Data:* Since both the objective function and its approximating function are defined on a space $\mathcal{S}$, one problem is that the training set $\mathbb{X}$ should be a reasonable sampling of the space $\mathcal{S}$, which directly relates to the fundamental assumption of machine learning that the training set has an identical distribution as the testing set has.
4) *Model Knowledge Parameters:* Different from the parameters inside the model that are acquired directly

from the training process, model knowledge parameters do not explicitly appear in the model, which are usually evaluated after the training process. For example, the uncertainty of classifier's outputs is a typical model knowledge parameter. The relationship between generalization and uncertainty of a classifier is initially demonstrated in [20]. This paper will conduct further studies on this relationship through incorporating a new index, i.e., complexity of classification.

B. Fuzziness of Classifier's Outputs

In this paper, we use fuzziness to depict the uncertainty of a classifier's outputs. The term "fuzziness," in conjunction with the concept of fuzzy set, was first mentioned by Zadeh [37]. He also generalized a probability measure of events that cannot be described by sharply defined collection of points, and suggested using entropy in information theory to interpret the uncertainty associated with a fuzzy event. De Luca and Termini [38] for the first time clearly proposed three properties that a fuzziness measure should satisfy. The term fuzziness can be interchangeable with "ambiguity" in some scenarios. Klir *et al.* [39], [40] stated that fuzziness and ambiguity gave two cognitive uncertainty measures.

As stated in [41], the fuzziness of a fuzzy set μ can be measured by a mapping $E(\mu)$:$F(\mathcal{S}) \rightarrow [0, \infty]$ where $F(\mathcal{S})$ denotes the space of all fuzzy sets defined on $\mathcal{S}$, satisfying the following axioms.

1) $E(\mu) = 0$ if and only if μ is a crisp set.
2) $E(\mu)$ attains its maximum value if and only if $\forall \mathbf{x} \in \mathcal{S}$: $\mu(\mathbf{x}) = 0.5$.
3) If $\mu \leq_s \sigma$, then $E(\mu) \geq E(\sigma)$, where $\leq_s$ is defined as

$$\mu \leq_s \sigma \Leftrightarrow \min(0.5, \mu(\mathbf{x})) \geq \min(0.5, \sigma(\mathbf{x})) \quad \max(0.5, \mu(\mathbf{x})) \leq \max(0.5, \sigma(\mathbf{x})).$$

4) $E(\mu) = E(\mu')$ when $\forall \mathbf{x} \in \mathcal{S}$: $\mu'(\mathbf{x}) = 1 - \mu(\mathbf{x})$.
5) $E(\mu \cup \sigma) + E(\mu \cap \sigma) = E(\mu) + E(\sigma)$.

Based on these axioms, we further introduce the following definition.

Definition 2 [32]: Let $B = \{\mu_1, \mu_2, \ldots, \mu_m\}$ be a fuzzy set, the fuzziness of B can be defined as

$$E(B) = -\frac{1}{m} \sum_{i=1}^{m} \big(\mu_i \log \mu_i + (1 - \mu_i) \log(1 - \mu_i)\big). \quad (7)$$

It is easy to verify that formula (7) indeed satisfies axioms 1–5.

Given a set of samples $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \subset \mathcal{R}^n \times \{0, 1\}^c$ and a well-trained classifier, a membership matrix $\mathbf{U} = [\mu_{ij}]$ can be obtained by matching each sample to the classifier, where $\mu_{ij} = \mu_j(\mathbf{x}_i)$ denotes the membership degree of the ith sample belonging to the jth class, where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, c$. It is worth noting that each output vector may not be a probability distribution, i.e., $\mu_{ij} \in [0, 1]$, and the equality $\sum_{j=1}^{c} \mu_{ij} = 1$ does not necessarily hold.

Based on Definition 2, the fuzziness of the classifier's outputs for the ith sample can be expressed as

$$E(\mu_i) = -\frac{1}{c} \sum_{j=1}^{c} \big(\mu_{ij} \log \mu_{ij} + \big(1 - \mu_{ij}\big) \log(1 - \mu_{ij})\big). \quad (8)$$

Having the above preliminaries, in the following, we propose a new concept to describe the fuzziness of a classifier's outputs on the entire training set.

Definition 3 (Fuzziness of a Classifier's Outputs): Suppose that a classifier is trained from training set $\mathbb{X}$. Without loss of generality, $\mathbb{X}$ is assumed to be a sufficient sampling of the entire sample space. Let $\mathbf{U} = [\mu_{ij}]_{c \times N}$ be the membership matrix given by matching each training sample to the classifier, where c is the number of classes and N is the number of samples. Then the fuzziness of the classifier's outputs can be defined as

$$E(\mathbf{U}) = -\frac{1}{cN} \sum_{i=1}^{N} \sum_{j=1}^{c} \big(\mu_{ij} \log \mu_{ij} + \big(1 - \mu_{ij}\big) \log\big(1 - \mu_{ij}\big)\big). \quad (9)$$

It is noted that Definition 3 uses the fuzziness of the classifier's outputs on the training set. In a more rigorous manner, it should be defined as the fuzziness of the classifier on the whole space. Unfortunately, the fuzziness of the classifier on unseen samples is unknown. According to the fundamental assumption of supervised learning that the training set is a reasonable and sufficient sampling of the entire sample space, we can use the classifier's fuzziness on the training set to approximately replace the classifier's fuzziness on the entire sample space.

C. Relationship Between Generalization and Fuzziness

Previous study [20] shows that the classifier with higher fuzziness of outputs has a better generalization for complex boundary problems when the training accuracy attains a predefined threshold. Furthermore, it demonstrates that the outputs of boundary samples have higher fuzziness, and samples with higher fuzziness exhibit higher risk of misclassification. By separating samples with high fuzziness from samples with low fuzziness, a divide-and-conquer learning algorithm based on fuzziness categorization was proposed in [41]. It shows that the category of sample with low or high fuzziness plays a critical role for performance improvement. Although these studies confirm that a relationship between fuzziness and generalization of a classifier indeed exists, it is difficult to explicitly express this relationship in general.

In the following, we make an investigation on data set *Spam*, which is a binary classification data set selected from UCI machine learning repository. This data set contains 4601 samples with 57 features. We randomly split it into two parts, i.e., 70% for training and 30% for testing. ELM is used to construct a classifier, which generates four indexes, i.e., training accuracy, testing accuracy, training fuzziness, and testing fuzziness. The random splitting is repeated for 100 times and four indexes are recorded for each repetition.

We make a statistical analysis for the 100 results. First, we split the interval between the minimum and maximum fuzziness values into ten parts with equal length and generate ten

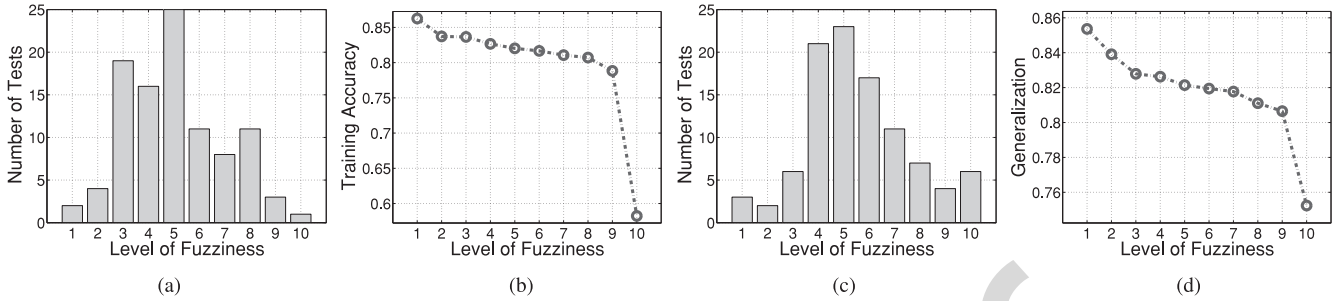


Fig. 1. Dependency relation between fuzziness and accuracy for *Spam*. (a) Histogram of training fuzziness. (b) Training accuracy. (c) Histogram of testing fuzziness. (d) Testing accuracy.

levels of fuzziness. For instance, the minimum and maximum fuzziness values for testing are 0.4889 and 0.5798, respectively. Then, the ten fuzziness levels for testing are generated as level 1 = [0.4889, 0.4980), level 2 = [0.4980, 0.5071), level 3 = [0.5071, 0.5162),..., and level 10 = [0.5707,0.5798]. Afterwards, we make a statistic for the number of experimental trials in each fuzziness level, and plot the histograms as shown in Fig. 1(a) and (c). Finally, we get the average training or testing accuracy for each fuzziness level, and plot the changing trends as shown in Fig. 1(b) and (d).

One can see from Fig. 1 that the relationship between accuracy and fuzziness of ELM does exist for *Spam*. We further calculate the Pearson correlation coefficient. As a remark, Pearson correlation reflects the statistical relationship between two sets of variables with a coefficient from $[-1, 1]$. A positive/negative coefficient represents that the two sets of variables are positive/negative correlated, and the absolute value represents the correlation degree. We use the median to represent each fuzziness level. Taking the testing result as an example, the correlation coefficient is calculated between fuzziness vector [0.4935, 0.5025, 0.5116, 0.5207, 0.5298, 0.5389, 0.5480, 0.5571, 0.5662, 0.5753] and accuracy vector [0.8536, 0.8391, 0.8279, 0.8263, 0.8214, 0.8194, 0.8177, 0.8111, 0.8065, 0.7524]. Finally, the correlation coefficients for training and testing are calculated as -0.7145 and -0.8625, respectively. This tells that the accuracy and fuzziness have a negative correlation for *Spam*, i.e., a higher fuzziness will lead to a lower accuracy, and the correlation degree is high.

Although the above example demonstrates that the relationship between generalization and uncertainty does exist for data set *Spam*, this relationship is difficult to express explicitly for general cases. In the subsequent sections, we will attempt to make this relationship clear by incorporating a new index, i.e., complexity of classification.

IV. Complexity of Classification Problem

Generally, a classification problem can be described as follows. Let $\mathcal{S}$ be the universal space we consider, F be a discrete function defined on $\mathcal{S}$. For simplicity, we suppose that function F takes values either 0 or 1, where 0 denotes one class and 1 denotes the other class. Given a subset of $\mathcal{S}$, denoted as $\mathbb{X}$, which is called the training set, the values of F on $\mathbb{X}$ are known, but the values of F on $\mathcal{S} - \mathbb{X}$ are unknown. A classification problem is to find a function f such that f can well approximate F both in $\mathbb{X}$ and $\mathcal{S} - \mathbb{X}$. Usually, F is called an objective function, f is called a classifier acquired based on training set $\mathbb{X}$, the approximation error on $\mathbb{X}$ is called training error, and the approximation error on $\mathcal{S} - \mathbb{X}$ represents the generalization ability of F.

The complexity of a classification problem refers to the complexity of function F, which implies the difficulties of the process of finding a quality f from $\mathbb{X}$. Unfortunately, there is no formal definition on the complexity of a discrete function. From references we can find a number of indexes to describe the complexity from different angles. It is noteworthy that the complexity of objective function is independent on the learned classifier f. Since the objective function F is unknown in real applications but is known on the training set $\mathbb{X}$, the indexes in describing the complexity of F can be estimated through the training set $\mathbb{X}$ and values of F on $\mathbb{X}$. In the following, we give several indexes to describe the complexity of F, which are mainly chosen from [21].

A. Fisher's Discriminant Ratio

Fisher's discriminant ratio is an old statistical index for describing the difference between two populations. Suppose that μ_{1j}, μ_{2j}, σ_{1j}, and σ_{2j} are the means and variances of the two populations (classes) with respect to the jth attribute, $j = 1, \ldots, n$. Then, the Fisher's discriminant ratio for the jth attributes is defined as

$$\mathfrak{f}_j = \frac{\left(\mu_{1j} - \mu_{2j}\right)^2}{\sigma_{1j}^2 + \sigma_{2j}^2}. \tag{10}$$

It is easy to see that Fisher's discriminant ratio with respect to the jth attribute describes the distance between two classes regarding this attribute. Intuitively, the longer the distance is, the easier the classification problem is, the lower the complexity will be. Thus, the complexity evaluating index is defined as

$$\mathfrak{Comp}_1 = \frac{1}{\max_j\{\mathfrak{f}_j\}}. \tag{11}$$

B. Volume of Overlap Region

A similar measure is the volume of overlap region between two class conditional distributions. It depends on, for each attribute, the maximum and the minimum values of each class.

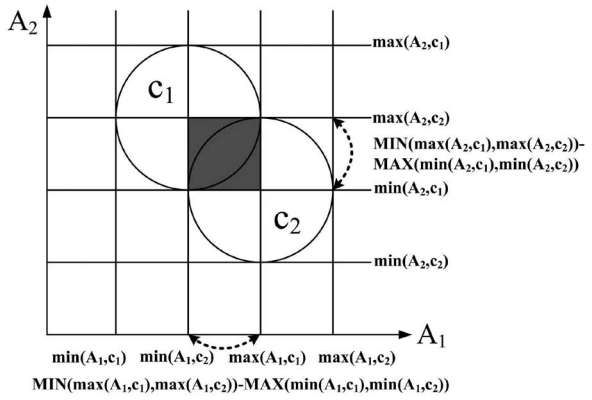


Fig. 2. Intuitive illustration of volume of overlap region.

We denote A_j as the jth attribute. Then, the overlap region normalized by the range of the value spanned by both classes, for each attribute A_j, can be represented as

$$\mathfrak{v}_j = \frac{\mathrm{MIN}\big(\max(A_j, c_1), \max(A_j, c_2)\big) - \mathrm{MAX}\big(\min(A_j, c_1), \min(A_j, c_2)\big)}{\mathrm{MAX}\big(\max(A_j, c_1), \max(A_j, c_2)\big) - \mathrm{MIN}\big(\min(A_j, c_1), \min(A_j, c_2)\big)} \tag{12}$$

where $\max(A_j, c_1)$, $\max(A_j, c_2)$, $\min(A_j, c_1)$, and $\min(A_j, c_2)$ denotes the maximum and minimum values of attribute A_j in the two classes, respectively. Then, the complexity evaluating index is defined as the volume of overlap region incorporating all the attributes

$$\mathfrak{Comp}_2 = \prod_{j=1}^{n} \mathfrak{v}_j \tag{13}$$

where n is the number of attributes. An intuitive illustration of volume of overlap region for a 2-D feature space is given in Fig. 2. It is noted that $\mathfrak{Comp}_2 = 0$ if the value ranges of the two classes do not overlap in at least one dimension. Obviously, a larger value of $\mathfrak{Comp}_2$ represents a higher complexity of the classification problem.

C. Intraclass/Interclass Distance Ratio

This measure first computes the Euclidean distance from each sample to its nearest neighbor within or outside the class. Assume that d_i^{intra} or d_i^{inter} is the distance between sample $\mathbf{x}_i$ and its nearest neighbor within or outside the class, we have

$$\begin{cases} d_i^{\mathrm{intra}} = \min_{j \neq i, y_j = y_i} d(\mathbf{x}_i, \mathbf{x}_j) \\ d_i^{\mathrm{inter}} = \min_{j \neq i, y_j \neq y_i} d(\mathbf{x}_i, \mathbf{x}_j) \end{cases} \tag{14}$$

where y_i and y_j represent the class labels of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Then, it takes the average of all the intraclass distances and the average of all the interclass distances, and the ratio of both averages is defined as the complexity of the problem

$$\mathfrak{Comp}_3 = \frac{\sum_{i=1}^{N} d_i^{\mathrm{intra}}}{\sum_{i=1}^{N} d_i^{\mathrm{inter}}} \tag{15}$$

where N is the number of samples. Similarly, a larger value of $\mathfrak{Comp}_3$ represents a higher complexity of the classification problem.

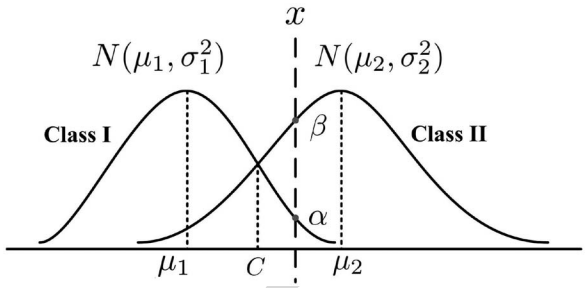


Fig. 3. Two normal populations.

D. Linear Separability

Linear separability was intensively discussed in the early literature. A simple definition to describe the linear separability for both separable and nonseparable cases is given by Smith [42]

$$\min \ \mathbf{a}^{\mathrm{T}}\mathbf{t}, \quad \text{s.t. } \mathbf{Z}^{\mathrm{T}}\mathbf{w} = \mathbf{t} \geq \mathbf{b} \tag{16}$$

where $\mathbf{a}$ and $\mathbf{b}$ are arbitrary constant vectors, $\mathbf{w}$ is the weight vector, $\mathbf{t} \geq 0$ is the error vector, and $\mathbf{Z}$ is a matrix in which each column $\mathbf{z}$ is defined based on the input vector $\mathbf{x}$ and its class label $\mathfrak{c}$

$$\begin{cases} \mathbf{z} = +\mathbf{x} & \text{if } \mathfrak{c} = c_1 \\ \mathbf{z} = -\mathbf{x} & \text{if } \mathfrak{c} = c_2. \end{cases} \tag{17}$$

The value of the objective function denotes the degree of being separable for two class cases, that is

$$\mathfrak{Comp}_4 = \mathbf{a}^{\mathrm{T}}\mathbf{t}. \tag{18}$$

It is noted that $\mathfrak{Comp}_4 = 0$ if the problem is linear separable.

Other indexes to describe the complexity of classification problem can be found from [21].

V. Relationship Between Generalization and Uncertainty by Incorporating Complexity of Classification

In this section, we give an analysis on the relationship between generalization and uncertainty by incorporating the complexity of classification. Since it is difficult for us to give a general analysis for all the complexity indexes, we only adopt the index of Fisher's discriminant ratio in Section IV-A, and give an explanation from the viewpoint of discriminant analysis, which has the principal of maximum probability.

Without loss of generality, we consider the 1-D case, which can be easily extended to multiple-dimensional cases. A normal distribution with mean μ and variance σ^2, denoted by $N(\mu, \sigma^2)$, has a probability density function

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \ -\infty < x < +\infty. \tag{19}$$

Suppose that there are two normal populations denoted by $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ as shown in Fig. 3, and $x(\mu_1 < x < \mu_2)$ is a new sample that needs to be discriminated.

For a classification problem, each population represents a class. From traditional textbook [43] we can view a simple way to judge sample x belonging to which class.

Let C be the cross-point between two density functions, i.e., C satisfies the following equation:

$$\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{(C-\mu_1)^2}{2\sigma_1^2}\right)=\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{(C-\mu_2)^2}{2\sigma_2^2}\right). \tag{20}$$

It is easy to check that the cross-point locates in the interval (μ_1, μ_2). The probabilities of sample x belonging to the two classes, denoted as (α, β), can be approximately viewed as

$$(\alpha,\beta)=\left(\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right),\ \frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)\right) \tag{21}$$

which induces the following discriminant rules based on the principle of maximum probability.

1) IF $x < C$ ($\alpha > \beta$) THEN x belongs to class I.
2) IF $x > C$ ($\alpha < \beta$) THEN x belongs to class II.
3) IF $x = C$ ($\alpha = \beta$) THEN the class of x is uncertain.

We now relate these discussions about discriminant analysis to the theme of this paper, i.e., uncertainty and complexity of a classification problem. According to Section IV-A, the complexity of a classification problem can be described by means and variances of class distributions. It can be roughly summarized as: the complexity is going up with either increasing the variances (σ_1^2, σ_2^2) or decreasing the difference between both means $|\mu_1 - \mu_2|$. Moreover, the uncertainty of a classifier is evaluated based on the probability vector (α, β) defined in (21). According to Section III, there are many specific formulas to evaluate the uncertainty (e.g., the fuzziness in Definition 3), but all of them have to satisfy the conditions given in Section III-B, e.g., if $\alpha < \beta$, when $\alpha' < \alpha$ and $\beta' > \beta$, the uncertainty output by vector (α', β') should be smaller than that output by (α, β). It shows that, to some extent, the difference between the two probability values denotes the magnitude of uncertainty. The bigger the difference is, the smaller the uncertainty is. Based on these analyses, we have the following theorems.

Theorem 1: Let

$$g(\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\left(\exp\left(-\frac{(x-\mu_2)^2}{2\sigma^2}\right)-\exp\left(-\frac{(x-\mu_1)^2}{2\sigma^2}\right)\right)$$

where $\sigma > 0$, $\mu_1 < \mu_2$, $x \in ([(\mu_1 + \mu_2)/2], \mu_2)$, and μ_1, μ_2 are considered as constants. Then, there exists a number $\sigma_1 \in (0, \mu_2 - x)$ such that $g(\sigma)$ is monotonically decreasing in the interval $(\sigma_1, +\infty)$.

Proof: The proof of Theorem 1 is listed in Appendix B. ■

Theorem 2: Let

$$q(\delta)=\frac{1}{\sqrt{2\pi}}\left(\exp\left(-\frac{x-(\mu_2-\delta)^2}{2}\right)-\exp\left(-\frac{x-(\mu_1+\delta^*)^2}{2}\right)\right)$$

where x, μ_1, and μ_2 are considered as constants, $\mu_1 < \mu_2$, $\delta^* = |[(\mu_1 - x)/(\mu_2 - x)]|\delta$, and $\delta > 0$. Then, there exists a number δ_1 such that $q(\delta)$ is monotonically decreasing in the interval $(0, \delta_1)$.

Proof: The proof of Theorem 2 can be derived similarly to the proof of Theorem 1. ■

Theorem 3: Suppose that the conditional probability outputs of a binary classifier follow two normal distributions $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$, respectively, where $\mu_1 < \mu_2$. Let

$$\alpha=-\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu_1)^2}{2\sigma^2}\right)\beta=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma^2}\right)$$

and

$$E(\alpha,\beta)=-\frac{1}{2}(\alpha\log\alpha+(1-\alpha)\log(1-\alpha)+\beta\log\beta+(1-\beta)\log(1-\beta)).$$

Assume $\beta = K\alpha$ where $K \in (1, 1+\epsilon)$, then $E(\alpha, \beta) = E(K)$ is monotonically decreasing with respect to K if $K\alpha > (1/2)$.

Proof: The proof of Theorem 3 is listed in Appendix C. ■

Noting that $g(\sigma)$ in Theorem 1 or $q(\delta)$ in Theorem 2 denotes the difference between two probability density values, which can be represented as $\beta - \alpha$ in Theorem 3. Theorem 3 directly connects this difference together with the uncertainty of the classifier's outputs given in Definition 2.

Theorem 3 shows that the uncertainty of the classifier's outputs is decreasing with the increase of the difference between two density values, i.e., $\beta - \alpha$, where α and β can be considered as the probabilities of a sample being classified as classes I and II, respectively. As a result, the conclusions in Theorems 1 and 2 show that the uncertainty of a classifier's outputs is becoming bigger with the increase of the complexity of the classification problem, which is represented through inflating the variance in Theorem 1 and through shrinking the difference between two means in Theorem 2, respectively. Since in a classification problem, the complexity is inherent while the uncertainty is generated by the output of a well-trained classifier which has its training and testing accuracy, it is reasonable to believe that some relationships exist among the accuracy, uncertainty, and complexity.

It is noteworthy that Theorems 1–3 cannot exactly explain the relationships among the three indexes, i.e., accuracy, uncertainty, and complexity. However, to a great extent, they provide solid supports to the existence of the relationships. They confirm such a fact that the classifier's uncertainty will be inevitably high if the classification problem is complex, no matter what classifier design algorithm is used. This statement further implies that a high-performance classifier will have high uncertainty when the problem is complex.

VI. Empirical Studies

In this section, we will conduct some empirical studies to further analyze the relationships discussed in Section V. It is noteworthy the discussions in Section V were made based

TABLE I
SELECTED DATA SETS FOR EXPERIMENTS

No	Data Set	# Samples	# Features	# Classes
1	Libras	369	90	15
2	Breast	699	9	2
3	SPECTF	267	44	2
4	Cancer	683	9	2
5	Chart	600	60	6
6	Cotton	356	21	6
7	CT	221	36	2
8	Dermatology	366	34	6
9	Ecoli	336	7	8
10	German	1,000	24	2
11	Glass	214	9	6
12	Haberman	306	3	2
13	Heart	270	13	2
14	Vowel	990	10	11
15	Ionosphere	351	34	2
16	Australian	690	14	2
17	Pima	768	8	2
18	Plrx	182	12	2
19	Sonar	208	60	2
20	Soybean	683	35	19
21	Bupa	345	6	2
22	Transfusion	748	4	2
23	Segment	2,310	19	7
24	Wdbc	569	30	2
25	Wpbc	198	33	2
26	Yeast	1,484	8	10
27	Zoo	101	16	7
28	Spam	4,601	57	2
29	Satellite	6,435	36	6
30	OptDigits	5,620	64	10
31	Pen	10,992	16	10

on $\mathfrak{Comp}_1$, i.e., Fisher's discriminant ratio. Thus, in this section, we will also adopt $\mathfrak{Comp}_1$ to evaluate the complexity of classification problems.

A. Selected Data Sets

The data sets used in the experiments are selected from UCI machine learning repository. The detailed information regarding these data sets are summarized in Table I. Since the complexity indexes listed in Section IV are defined for binary classification problems, we transfer each multiclass data set into binary by randomly selecting 50% classes as positive and the rest 50% classes as negative.

B. Experimental Design

The flowchart for training the classifier and evaluating the problem complexity is listed in Algorithm 1.

It is noteworthy that the training algorithm adopted in this section is ELM. Due to the random mechanism for weight assignment, it is easy to repeat the experiment for many times. We conduct 100 experimental trials for each data set. In each trial, 70% data are randomly selected for training, and the remaining 30% data are used for testing. Each trial will provide a different result, and we make statistics for fuzziness, accuracy, and complexity based on the 100 results.

The number of hidden nodes in ELM is set as 20, and sigmoid activation function is utilized. The simulations are carried out under MATLAB R2011b, which are executed on a computer with an Intel Core i7-5500U CPU@2.40 GHz, 8GB memory, and 64-bit Windows 8 system.

Algorithm 1: Train ELM Classifier and Compute Evaluating Indexes

Input:
Training set $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \subset \mathcal{R}^n \times \{0, 1\}^c$;
Activation function $f(\mathbf{x})$;
Number of hidden nodes $\tilde{N}$.

Output:
Fuzziness and generalization of the trained classifier;
Complexity of the classification problem.

1 *Data processing*: randomly divide the data set into two parts for training and testing according to a separation ratio.
2 *Classifier training*: train a ELM classifier based on the algorithm given in section II-A.
3 *Testing*: test the classifier on the testing set, compute the fuzziness (Definition 3) and generalization (testing accuracy) of the classifier.
4 *Complexity evaluation*: compute the complexity of the classification problem, i.e., Eq. (11).

C. Experimental Analysis

Similar to Section III-C, we make some statistical analyses on the testing results. For each data set, ten fuzziness levels are generated by equally dividing the interval between the maximum and minimum fuzziness values. We use the median to represent each fuzziness level. Then, the number of experimental trials for each fuzziness level is counted, and the average testing accuracy for each fuzziness level is calculated. Fig. 4 demonstrates the changing trend of the testing accuracy along with the level of fuzziness. It depicts the dependency relation between testing accuracy and testing fuzziness for the classification problems. Due to space limit, we only plot the results for 12 data sets out of 31. Furthermore, we calculate the Pearson correlation coefficient between fuzziness vector and accuracy vector for each data set. It is noteworthy there are ten fuzziness levels for each data set. However, from Fig. 4, we can see that the highest fuzziness level (i.e., level ten) usually cause a sharp change of the testing accuracy, which may interfere the statistical analysis for the overall results. Thus, we only use the previous nine fuzziness values and their corresponding accuracy. The correlation coefficients r are listed in Table II. We artificially set up some thresholds to justify the degree of correlation.

1) If $0 \leq |r| < 0.4$, then the correlation is low.
2) If $0.4 \leq |r| < 0.7$, then the correlation is medium.
3) If $0.7 \leq |r| \leq 1$, then the correlation is strong.

It is observed from Table II that the generalization and fuzziness have a strong or medium correlation regarding most data sets.

The complexities of the problems are shown in Fig. 5, which are sorted according to the order numbers (i.e., 1–31) in Table I. In Fig. 5, we artificially set up a threshold such that the complexity higher than the threshold is called high otherwise is called low. In this case, one can view an implicit relation among the complexity, generalization, and fuzziness.

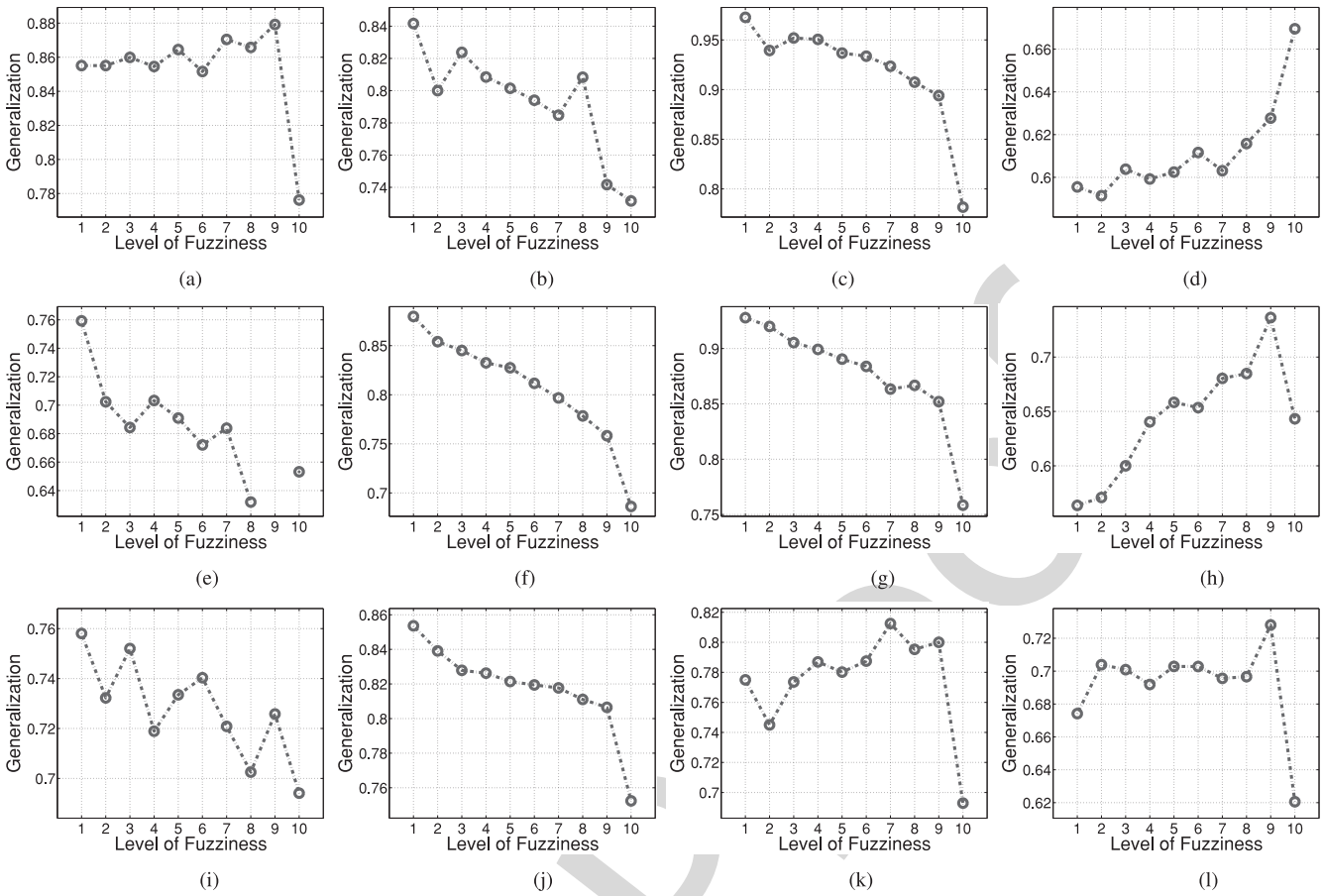


Fig. 4. Relationship between fuzziness and generalization of ELM classifier on different data sets. (a) Australian. (b) Chart. (c) Dermatology. (d) Segment. (e) Libras. (f) OptDigits. (g) Pen. (h) Plrx. (i) Sonar. (j) Spam. (k) SPECTF. (l) Yeast.

TABLE II
PEARSON CORRELATION COEFFICIENT BETWEEN OUTPUT FUZZINESS AND TESTING ACCURACY

Data Set	Pearson Correlation Coefficient	Data Set	Pearson Correlation Coefficient
1	-0.6434√	17	-0.0520†
2	-0.3522†	18	0.9743★
3	0.7838★	19	-0.6896√
4	-0.7835★	20	-0.9348★
5	-0.7718★	21	0.4132√
6	0.3421†	22	0.4962√
7	0.3744†	23	0.8728★
8	-0.9277★	24	-0.2933†
9	-0.0579†	25	-0.5559√
10	-0.1474†	26	0.6297√
11	-0.5452√	27	0.3470†
12	0.5803√	28	-0.9496★
13	0.1768†	29	0.1455†
14	-0.9362★	30	-0.9903★
15	0.5782√	31	-0.9895★
16	0.7420★		

Note: For each data set, ★ represents strong correlation, √ represents medium correlation, and † represents low correlation.

The generalization of a classifier trained by ELM goes up with the increase of fuzziness if the complexity of the classification problem is relatively high, while the generalization of a classifier trained by ELM goes down with the increase of fuzziness if the complexity of the classification problem is relatively low. For instance, it can be seen from Fig. 5 that the complexity values of *Segment* (data set 23) and *Plrx* (data set 18) are high, in this case, the generalizations of these two data sets are becoming better with the increase of fuzziness as shown in Fig. 4(d) and (h). However, the complexity values of *OptDigits* (data set 30) and *Spam* (data set 28) are low, in this case, the generalizations of these two data sets are becoming worse with the increase of fuzziness as shown in Fig. 4(f) and (j).

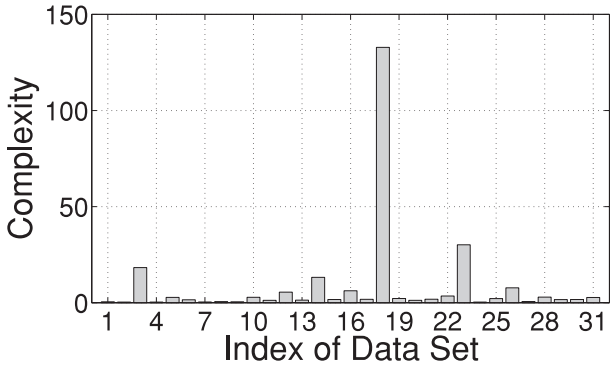


Fig. 5. Complexity of the classification problems.

By learning the complexity of classification problems from Fig. 5, we grasp some factors that are resulted from the

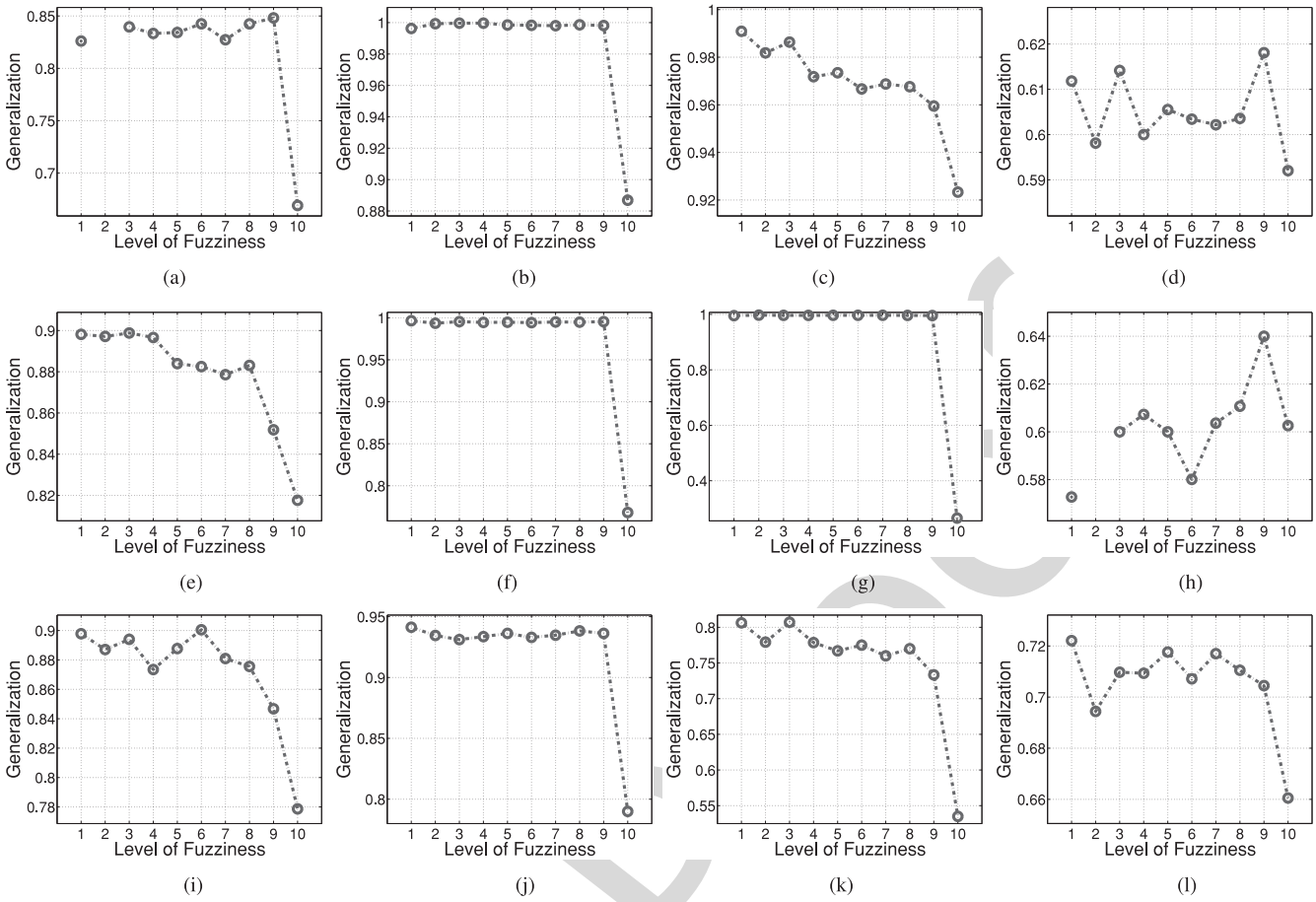


Fig. 6. Relationship between fuzziness and generalization of SVM classifier on different data sets. (a) Australian. (b) Chart. (c) Dermatology. (d) Segment. (e) Libras. (f) OptDigits. (g) Pen. (h) Plrx. (i) Sonar. (j) Spam. (k) SPECTF. (l) Yeast.

complexity of decision boundaries. It is obvious that there are some relations between them.

As we know, the complexity of a classification problem can be intuitively regarded as the degree of difficulty for the problem. More specifically, it is the complexity of geometrical class boundary which can be seen as an equation $F = 0$ that divides the sample space. In classification problem, it is desired to find a classifier f by training the data set locating next to the boundary function $F = 0$. The ability of function f to approximate function F on unseen data is the generalization, and the fuzziness of the classifier is the uncertainty of function f in dividing unseen samples.

When it is easy to distinguish the classes by the boundary of function F, it will also be easy to divide the unseen samples by f, since the structure of training data is supposed to be similar to the structure of unseen data and f is an estimator of F. It implies that the boundary will be simple and the fuzziness of the boundary is low. In this situation, it is reasonable to believe that, with the decrease of classifier's fuzziness, the generalization will be improved.

When it is difficult to distinguish the classes by the boundary of function F, the classifier function f is also difficult to divide the unseen samples. It corresponds to a case of high complexity and complex boundary. It is inherent to output high fuzziness for boundary samples for any classifier, and therefore, we reasonably believe in this situation that, with the increase of classifier's fuzziness, the generalization may be getting better.

D. Analysis With SVM Classifiers

We further realize the above studies with SVM classifiers. We adopt the "LibSVM" toolbox, the penalty term C is fixed as 100, and RBF kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}_i) = \exp(-[||\mathbf{x} - \mathbf{x}_i||^2/2\sigma^2])$ with $\sigma = 1$ is adopted. The decision values of SVM are transformed into uncertain outputs by logistic function. The dependency relation between generalization and fuzziness regarding the 12 data sets in Fig. 4 are demonstrated in Fig. 6. It can be observed that the results are basically consistent with those in Section VI-C, but the changing trends are not as clear as those of ELM. As a result, ELM might be more suitable to conduct this paper, since it has a higher degree of uncertainty due to the random mechanism for input weights assignment.

VII. Conclusion

This paper finds an empirical relationship among the complexity of a classification problem, the uncertainty of classifier's outputs, and the prediction accuracy of the classifier. By experimental validation and theoretical explanation through a simple model of discriminant analysis, it is found that with the

increase of the uncertainty of the classifier's outputs, empirically the accuracy is upgrading for high-complexity problem but downgrading for low-complexity problem. Based on these findings, in order to choose a better classification rule for a practical problem, one can tune the model parameters such that the uncertainty becomes larger for problems with higher complexity, or smaller for problems with lower complexity under the condition that an acceptable training accuracy is kept.

APPENDIX A
FEATURES OF ELMS

In the following, we briefly review the major advantages of ELMs.

1) The first advantage of ELMs is the fast training speed. Since the training of ELMs does not include iterative tuning, it statistically shows that ELM is thousands of times faster than BP given a predefined threshold for training accuracy.
2) Another feature of ELMs is the acceptable generalization ability. In comparison with other popular classification or regression algorithms, such as DTs, SVMs, logistic regressions, etc., the generalization of ELMs may not be the best in general. But so far, one cannot find a significant difference among the generalizations of these algorithms.
3) The training procedure of ELMs can process online sequential data conveniently, which demonstrates strong potentials for big data analytic. It is shown that ELMs can effectively handle both numerical and nominal attributes for both classification and regression problems.
4) Mathematically it is proven that ELMs have the universal approximation ability if the activation function is differentiable. That is, ELMs can uniformly approximate any continuous function defined in an interval when the number of hidden nodes goes to infinity. This conclusion establishes the foundation of applying ELMs to various classification and regression problems.

It is worthy noting that any learning algorithm cannot be consistently better than others. In the following, we list several disadvantages of ELMs.

1) As aforementioned, the weights between input and hidden layers in ELMs are randomly selected from an interval. ELMs are sensitive to this interval, and the change of the interval will produce quite different classifiers, which seriously decreases the stability.
2) The number of hidden layer nodes is critical for building an ELM. A large number will lead to the generalization decreasing but a small number can result in the training error increasing. So far, how to select the number of hidden layer nodes is still a challenging issue.

APPENDIX B
PROOF OF THEOREM 1

The original problem can be represented as

$$g(\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\left(\exp\left(-\frac{(x-b)^2}{2\sigma^2}\right) - \exp\left(-\frac{(x-a)^2}{2\sigma^2}\right)\right)$$

prove that there exits σ_1 such that $g(\sigma)$ is monotonically increasing when $\sigma < \sigma_1$ and $g(\sigma)$ is monotonically decreasing when $\sigma > \sigma_1$.

The constant term $\sqrt{2\pi}$ can be neglected. Let $(x-a) = k \times (b-x)$ and $\sigma = t \times (b-x)$, the original problem can be simplified as

$$g(t) = \frac{1}{t}\left(\exp(-\frac{1}{2t^2}) - \exp(-\frac{k^2}{2t^2})\right), \quad k > 1 \text{ and } t > 0$$

prove that there exits t_1 such that $g(t)$ is monotonically increasing when $t < t_1$ and $g(t)$ is monotonically decreasing when $t > t_1$.

We get the first-order derivation of $g(t)$, that is

$$g'(t) = \frac{1}{t^4}\left[\left(1-t^2\right)\exp\left(-\frac{1}{2t^2}\right) - \left(k^2-t^2\right)\exp\left(-\frac{k^2}{2t^2}\right)\right].$$

Having this derivation, it can be derived as follows.

1) When $t > k$, $t^2 - 1 > t^2 - k^2 > 0$ and $\exp(-[1/2t^2]) > \exp(-[k^2/2t^2])$, thus $(t^2-1)\exp(-[1/2t^2]) > (t^2-k^2)\exp(-[k^2/2t^2])$, thus we have $g'(t) < 0$.
2) When $k \geq t > 1$, $(1-t^2)\exp(-[1/2t^2]) < 0$, thus $(k^2-t^2)\exp(-[k^2/2t^2]) > 0$, thus we have $g'(t) < 0$.
3) When $t = 1$, we have $g'(t) = [1/t^4][-(k^2-t^2)\exp(-[k^2/2t^2])] < 0$.

So far, we have proved that $g'(t) < 0$ when $t \geq 1$, which means that $g(t)$ is monotonically decreasing when $t \geq 1$.

When $1 > t > 0$ and $t \to 0$, we have $[(1-t^2)/(k^2-t^2)] \to (1/k^2)$ and $\exp([(1-k^2)/2t^2]) \to 0$ (noting that $t \leq 1 < k$). There exists $t^* \in (0,1)$ such that $([(1-t^{*2})/(k^2-t^{*2})] > \exp([(1-k^2)/2t^{*2}]) = [\exp(1/2t^{*2})/\exp([k^2/2t^{*2}])]$, thus $([(1-t^{*2})\exp(-1/2t^{*2})]/[(k^2-t^{*2})\exp(-k^2/2t^{*2})]) > 1$, thus $(1-t^{*2})\exp(-1/2t^{*2}) > (k^2-t^{*2})\exp(-k^2/2t^{*2})$, thus $g'(t^*) > 0$.

According to Zero theorem, there exits $t_1 \in (0,1)$ such that $g'(t_1) = 0$. Since $g'(t)$ is continuous and differentiable, if all the stagnation points are maximum points, then there is only one stagnation point, otherwise minimum point exists.

We further get the second-order derivation of $g(t)$, that is

$$g''(t) = \frac{1}{t^7}\left\{\left[2t^2\left(t^2-1\right) - 2t^2 + (1-t^2)\right]\exp\left(-\frac{1}{2t^2}\right) - \left[2t^2\left(t^2-k^2\right) - 2t^2k^2 + k^2\left(k^2-t^2\right)\right]\exp\left(-\frac{k^2}{2t^2}\right)\right\}.$$

Put the stagnation point t_1 into $g''(t)$, since $(1-t_1^2)\exp(-1/2t_1^2) - (k^2-t_1^2)\exp(-k^2/2t_1^2) = 0$, we have

$$g''(t_1) = \frac{1}{t_1^7}\left\{-2t_1^2\left[\exp\left(-\frac{1}{2t_1^2}\right) - k^2\exp\left(-\frac{k^2}{2t_1^2}\right)\right] + \left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - k^2\left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right)\right\}.$$

Based on

$$\left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - \left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right) = 0$$

$$k > 1 \text{ and } 1 > t_1 > 0$$

we have

$$\exp\left(-\frac{1}{2t_1^2}\right) - k^2\exp\left(-\frac{k^2}{2t_1^2}\right) = t_1^2\left[\exp\left(-\frac{1}{2t_1^2}\right) - \exp\left(-\frac{k^2}{2t_1^2}\right)\right] > 0$$

and

$$\left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - k^2\left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right) < \left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - \left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right) = 0.$$

Thus, $g''(t_1) < 0$, t_1 is the maximum point, which means that $g(t)$ is monotonically increasing when $t < t_1$ and $g(t)$ is monotonically decreasing when $t_1 < t < 1$.

To this end, we have proved that $g(t)$ is monotonically increasing when $t < t_1$ and $g(t)$ is monotonically decreasing when $t > t_1$.

APPENDIX C
PROOF OF THEOREM 3

Substituting β with $K\alpha$ in $E(K)$, we have

$$E(K) = -\frac{1}{2}\big(\alpha\log\alpha + (1-\alpha)\log(1-\alpha) + K\alpha\log(K\alpha) + (1-K\alpha)\log(1-K\alpha)\big).$$

Taking derivative of $E(K)$ with respect to K, we obtain

$$\frac{dE(K)}{d(K)} = -\frac{1}{2}(\alpha\log(K\alpha) - \alpha\log(1-K\alpha)) = -\frac{\alpha}{2}\log\frac{K\alpha}{1-K\alpha}.$$

It is easy to view that $[dE(K)/d(K)] < 0$ if $K\alpha > (1/2)$, which completes the proof.

REFERENCES

[1] W. W. Y. Ng, A. P. F. Chan, D. S. Yeung, and E. C. C. Tsang, "Quantitative study on the generalization error of multiple classifier systems," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2005, pp. 889–894.

[2] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[3] X. D. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[4] Z. Yan and C. Xu, "Studies on classification models using decision boundaries," in *Proc. 8th IEEE Int. Conf. Cogn. Informat.*, Hong Kong, 2009, pp. 287–296.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2012.

[6] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 2000.

[7] R. Wang, S. Kwong, and D. Chen, "Inconsistency-based active learning for support vector machines," *Pattern Recognit.*, vol. 45, no. 10, pp. 3751–3767, 2012.

[8] R. Wang and S. Kwong, "Active learning with multi-criteria decision making systems," *Pattern Recognit.*, vol. 47, no. 9, pp. 3106–3119, 2014.

[9] R. Wang, D. Chen, and S. Kwong, "Fuzzy-rough-set-based active learning," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1699–1704, Dec. 2014.

[10] R. Wang, C.-Y. Chow, and S. Kwong, "Ambiguity-based multiclass active learning," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 242–248, Feb. 2016.

[11] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[12] R. Wang, S. Kwong, X.-Z. Wang, and Q. Jiang, "Segment based decision tree induction with continuous valued attributes," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1262–1275, Jul. 2015.

[13] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[14] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[15] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989.

[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[17] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[18] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 1, pp. 1–14, Feb. 1998.

[19] Z. Deng, K.-S. Choi, Y. Jiang, and S. Wang, "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2585–2599, Dec. 2014.

[20] X.-Z. Wang *et al.*, "A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1638–1654, Oct. 2015.

[21] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 289–300, Mar. 2002.

[22] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[23] E. Soria-Olivas *et al.*, "BELM: Bayesian extreme learning machine," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 505–509, Mar. 2011.

[24] G.-B. Huang, H. M. Zhou, X. J. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[25] A. A. Mohammed, R. Minhas, Q. M. J. Wu, and M. A. Sid-Ahmed, "Human face recognition based on multidimensional PCA and extreme learning machine," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2588–2597, 2011.

[26] K. A. Toh, "Deterministic neural classification," *Neural Comput.*, vol. 20, no. 6, pp. 1565–1595, 2008.

[27] X. Liu, S. Lin, J. Fang, and Z. Xu, "Is extreme learning machine feasible? A theoretical assessment (part I)," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 7–20, Jan. 2015.

[28] S. Lin, X. Liu, J. Fang, and Z. Xu, "Is extreme learning machine feasible? A theoretical assessment (part II)," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 21–34, Jan. 2015.

[29] J. Cao, K. Zhang, M. Luo, C. Yin, and X. Lai, "Extreme learning machine and adaptive sparse representation for image classification," *Neural Netw.*, vol. 81, no. C, pp. 91–102, Sep. 2016.

[30] J. Cao, J. Hao, X. Lai, C.-M. Vong, and M. Luo, "Ensemble extreme learning machine and sparse representation classification algorithm," *J. Franklin Inst.*, vol. 353, no. 17, pp. 4526–4541, 2016.

[31] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[32] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proc. IEEE Symp. Comput. Intell. Data Min.*, Nashville, TN, USA, 2009, pp. 389–395.

[33] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, nos. 1–3, pp. 359–366, 2008.

[34] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1352–1357, Aug. 2009.

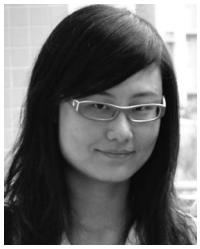
[35] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[36] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Budapest, Hungary, 2004, pp. 985–990.
[37] L. A. Zadeh, "Probability measures of fuzzy events," *J. Math. Anal. Appl.*, vol. 23, no. 2, pp. 421–427, 1968.
[38] A. De Luca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory," *Inf. Control*, vol. 20, no. 4, pp. 301–312, 1972.
[39] G. J. Klir, "Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?" *Fuzzy Set Syst.*, vol. 24, no. 2, pp. 141–160, 1987.
[40] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty and Information*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.
[41] D. Sánchez and E. Trillas, "Measures of fuzziness under different uses of fuzzy sets," in *Advances in Computational Intelligence* (Communications in Computer and Information Science), vol. 298. Heidelberg, Germany: Springer, 2012, pp. 25–34.
[42] F. W. Smith, "Pattern classifier design by linear programming," *IEEE Trans. Comput.*, vol. C-17, no. 4, pp. 367–372, Apr. 1968.
[43] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.
[44] S. Ding, N. Zhang, and J. Zhang, "Unsupervised extreme learning machine with representational features," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 2, pp. 587–595, Apr. 2017s.
[45] S. Balasundaram and D. Gupta, "On optimization based extreme learning machine in primal for regression and classification by functional iterative method," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 5, pp. 707–728, Oct. 2016.
[46] P. Liu, Y. Huang, L. Meng, and S. Gong, "Two-stage extreme learning machine for high-dimensional data," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 5, pp. 765–772, Oct. 2016.
[47] J. Zhang, S. Ding, and N. Zhang, "Incremental extreme learning machine based on deep feature embedded," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 1, pp. 111–120, Feb. 2016.
[48] A. Fu, C. Dong, and L. Wang, "An experimental study on stability and generalization of extreme learning machines," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 1, pp. 129–135, Feb. 2015.

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

From 2001 to 2014, he was a Full Professor and the Dean of the College of Mathematics and Computer Science, Hebei University, Hebei, China. From 1998 to 2001, he was a Research Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. Since 2014, he has been a Full Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include supervised and unsupervised learning, active learning, reinforcement learning, manifold learning, transfer learning, unstructured learning, uncertainty, fuzzy sets and systems, fuzzy measures and integrals, rough set, and learning from big data.

Dr. Wang was a recipient of many awards from the IEEE SMC Society. He is a member of the Board of Governors of the IEEE International Conference on Systems, Man, and Cybernetics (SMC) in 2005, 2007–2009, and 2012–2014, the Chair of the Technical Committee on Computational Intelligence of the IEEE SMC, and a Distinguished Lecturer of the IEEE SMC. He was the Program Co-Chair of the IEEE SMC 2009 and 2010. He is the Editor-in-Chief of the *International Journal of Machine Learning and Cybernetics*. He is also an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, *Information Sciences Journal*, and the *International Journal of Pattern Recognition and Artificial Intelligence*.



Ran Wang (S'09–M'14) received the B.Eng. degree in computer science from the College of Information Science and Technology, Beijing Forestry University, Beijing, China, in 2009, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2014.

From 2014 to 2016, she was a Post-Doctoral Researcher with the Department of Computer Science, City University of Hong Kong. She is currently an Assistant Professor with the College of Mathematics and Statistics, Shenzhen University, Shenzhen, China. Her current research interests include pattern recognition, machine learning, fuzzy sets and fuzzy logic, and their related applications.

Chen Xu received the B.Sc. and M.Sc. degrees from Xidian University, Xi'an, China, in 1986 and 1989, respectively, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, in 1992.

He joined Shenzhen University, Shenzhen, China, in 1992, where he is currently a Professor. From 1999 to 2000, he was a Research Fellow with Kansai University, Suita, Japan, and the University of Hawaii, Honolulu, HI, USA, from 2002 to 2003. His current research interests include image processing, intelligent computing, and wavelet analysis.

Discovering the Relationship Between Generalization and Uncertainty by Incorporating Complexity of Classification

Xi-Zhao Wang, *Fellow, IEEE*, Ran Wang, *Member, IEEE*, and Chen Xu

***Abstract*—The generalization ability of a classifier learned from a training set is usually dependent on the classifier's uncertainty, which is often described by the fuzziness of the classifier's outputs on the training set. Since the exact dependency relation between generalization and uncertainty of a classifier is quite complicated, it is difficult to clearly or explicitly express this relation in general. This paper shows a specific study on this relation from the viewpoint of complexity of classification by choosing extreme learning machines as the classification algorithms. It concludes that the generalization ability of a classifier is statistically becoming better with the increase of uncertainty when the complexity of the classification problem is relatively high, and the generalization ability is statistically becoming worse with the increase of uncertainty when the complexity is relatively low. This paper tries to provide some useful guidelines for improving the generalization ability of classifiers by adjusting uncertainty based on the problem complexity.**

***Index Terms*—Complexity of classification, extreme learning machine, generalization, uncertainty.**

I. Introduction

CLASSIFICATION problem, as the central part in the fields of pattern recognition and data mining, refers to a task of assigning objects to one of several predefined class labels. Given a set of objects, the mathematical model of classification problem is a discrete-valued function that maps each object to a class label. Usually, the process of determining the discrete-valued function from a training set is called learning while the process of using the determined function to classify a new object is called reasoning [1]–[5].

For a classification problem with c classes, the reasoning result is generally a c-dimensional vector. According to the output forms of the reasoning process, existing learning algorithms can be classified into two categories. In one category, the c-dimensional output vector contains one component of value 1 and other components of value 0. In this situation, the class label corresponding to the component 1 will be the reasoning result. This kind of algorithms are known as crisp-output algorithms, such as traditional support vector machine (SVM) [6]–[10], decision tree (DT) [11], [12], etc. In the other category, the c-dimensional output vector contains components of real values within the interval [0, 1]. In this situation, the class label corresponding to the maximum component will be the reasoning result. If the maximum is attained at more than one component, a special strategy will be designed to determine the final result. This kind of algorithms are acknowledged as uncertain-output algorithms, such as k-nearest neighbor [2], Bayesian probability model [2], back-propagation (BP) methods for training feed-forward neural networks [13]–[16], etc.

Obviously, crisp-output algorithms are special cases of uncertain-output algorithms. If an algorithm belongs to the crisp category, then it belongs to the uncertain category, however, it is not true conversely. Most crisp-output algorithms can be extended to uncertain-output algorithms, such as fuzzy SVM [17], fuzzy DT [18], etc. In this paper, we will intensively investigate the uncertain-output algorithms, which highlight the argument that uncertainty does exist in the learning and reasoning processes.

On the other hand, generalization of a classifier is defined as the rate of the correctly classified objects that are not in the training set. It is the most important index for evaluating a classification algorithm since the ultimate goal for developing a classification model is to achieve high prediction accuracy on unseen cases. Usually, the generalization of a classifier depends on multiple factors.

1) The mathematical model, which has a direct impact on both the training accuracy and testing accuracy.
2) The algorithm for training the model parameters, which is sensitive to the prediction results.

Manuscript received July 4, 2016; revised November 28, 2016; accepted January 11, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61402460, Grant 61472257, Grant 61170040, and Grant 71371063, in part by the Basic Research Project of Knowledge Innovation Program in Shenzhen under Grant JCYJ20150324140036825, in part by the Guangdong Provincial Science and Technology Plan Project under Grant 2013B040403005, and in part by the HD Video Research and Development Platform for Intelligent Analysis and Processing in Guangdong Engineering Technology Research Centre of Colleges and Universities under Grant GCZX-A1409. *(Corresponding author: Ran Wang.)*

X.-Z. Wang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: xizhaowang@ieee.org).

R. Wang and C. Xu are with the College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, China (e-mail: wangran@szu.edu.cn; xuchen@szu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2653223

3) *The data distribution:* In supervised learning, there is a fundamental assumption that the training data has the same distribution as the testing data. The learning scheme that does not follow this fundamental assumption is referred to as transfer learning [19], which is out of the scope of this paper.

Many research efforts have been made to improve the generalization of a classifier by considering different factors. In this paper, we consider a particular model parameter, i.e., the uncertainty of the classifier's outputs, which has been proven in [20] to have a close relationship with the generalization of classifier. It has been shown in [20] that the uncertainty of the classifier's outputs has a close relationship with the generalization capability. However, this relationship is difficult to express explicitly for general cases. In order to further investigate this relationship, in this paper, we take into account a new index, i.e., complexity of classification, which can be measured in different ways [21]. To the best of our knowledge, this paper makes a first attempt to investigate the relationship between generalization and uncertainty of a classifier by incorporating the complexity of classification.

In addition, choosing an appropriate classification algorithm is also an important issue to conduct this research. It is noteworthy that any uncertain-output algorithm can be used to study the relationship between generalization and uncertainty. As the commonly used classification model for various practical problems, feed-forward neural networks will be adopted. The most notable algorithm to train a feed-forward neural network is BP. Although it has been proved in [15] and [16] that BP network has the ability to approximate any continuous function with arbitrary precision, it is often criticized to have the problems of slow convergence speed and local minima. In order to overcome these deficiencies, extreme leaning machine (ELM) has been proposed as a new training algorithm for single-hidden layer feed-forward neural network (SLFN) [22]. Differentiating from BP that iteratively tunes the weight parameters by gradient descent technique, ELM randomly chooses the weight parameters between input and hidden layers and analytically solves the weight parameters between hidden and output layers through Moore–Penrose generalized inverse [44]–[48]. Due to the extremely fast training speed and good prediction performance, ELM has been investigated intensively and extensively in the machine learning and data mining communities [23]–[26]. Based on the aforementioned advantages, we will adopt ELM as the classification algorithm in this paper. The major theoretical issues of ELM can be found in [27] and [28], and the applications of ELM to different areas, such as sparse representation can be found in [29] and [30].

The rest of this paper is organized as follows. Section II reviews ELMs. Section III introduces the dependency relation between generalization and uncertainty of classifiers. Section IV discusses the complexity of classification problems. Section V analyzes the relationship between generalization and uncertainty by incorporating a complexity index. Experiments are conducted in Section VI. Finally, conclusions are given in Section VII.

II. Extreme Learning Machine

This section will introduce ELM, which is a noniterative training algorithm for SLFNs.

A. Training of ELM

A standard SLFN for classification is a discrete function mapping samples to class labels. Given a training set that contains N arbitrarily distinct samples $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \subset \mathcal{R}^n \times \{0, 1\}^c$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ is the ith training sample, $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{ic}]$ is the label vector of $\mathbf{x}_i$, n is the number of features, and c is the number of classes. An SLFN with $\widetilde{N}$ hidden nodes and activation function $g(\mathbf{x})$ can be expressed as

$$\sum_{j=1}^{\widetilde{N}} \beta_j g\left(\mathbf{w}_j \cdot \mathbf{x}_i + b_j\right) = \mathbf{t}_i, \ \ i = 1, 2, \ldots, N \tag{1}$$

where $\mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jn}]$ is the weight linking the input nodes to the jth hidden node, b_j is the bias of the jth hidden node, β_j is the weight linking the jth hidden node to the output nodes, and sigmoid function $g(x) = (1/[1 + \exp(-x)])$ is selected as the activation function.

In ELMs, the input weights $\mathbf{w}_j$ and biases b_j are randomly chosen, and the learning can be formulated as a minimum optimization problem with a regularized term

$$\min_\beta \left\{ ||\mathbf{T} - \mathbf{H}\beta||_2^2 + \mu ||\beta||_2^2 \right\}, \ \ \mu > 0 \tag{2}$$

where $\mathbf{H}$ is the hidden layer output matrix denoted as

$$\begin{aligned} &\mathbf{H}(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{\widetilde{N}}, b_1, b_2, \ldots, b_{\widetilde{N}}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \\ &= \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g\left(\mathbf{w}_{\widetilde{N}} \cdot \mathbf{x}_1 + b_1\right) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g\left(\mathbf{w}_{\widetilde{N}} \cdot \mathbf{x}_N + b_{\widetilde{N}}\right) \end{bmatrix}_{N \times \widetilde{N}} \end{aligned} \tag{3}$$

and $\mathbf{T}$ is the label matrix denoted as

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{bmatrix}_{N \times c}. \tag{4}$$

The optimal estimation of output weights β^* can be formulated as a regularized least square problem

$$\beta^*_{\widetilde{N} \times c} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \mu\mathbf{I}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T} \tag{5}$$

where $\mathbf{I}$ is the identity matrix of suitable dimension and μ is the regularizing factor.

To this end, all the parameters $\{\mathbf{w}, b, \beta\}$ in ELM have been fixed, and the training process is finished.

ELMs have been proved to have the universal approximation capabilities [31] although the training process does not include any iteration. Under the assumption of smoothness of the underlying function, the universal approximation capability of ELMs can be guaranteed by providing a sufficiently large number of hidden nodes with certain range of $\mathbf{w}$ and b.

In comparison with BP algorithm, ELMs have a much faster training speed due to the noniterative mechanism. References show that ELMs can finish the training process thousands of times faster than BP in some scenarios, at the same time, an

acceptable learning accuracy is kept. The advantages and disadvantages of ELMs are listed in Appendix A. Furthermore, one can find many improved versions for ELMs. The computation of weights between hidden and output layers can be improved through an optimization algorithm given by Deng *et al.* [32] in order to avoid over-fitting. Rong *et al.* [33] offered a pruned ELM in which the corresponding nodes can be removed according to the information gain to reduce the correlation among classes in a large network structure. Feng *et al.* [34] proposed an EM-ELM in which the weights are not updated when a node is added, and the algorithm can update the weights and adjust the network at the same time. Furthermore, it is found that ELMs can online deal with sequential data successfully [35].

B. Generalized Inverse and Normal Equations

In ELMs, the weights between hidden and output layers are calculated by the generalized inverse [36]. We briefly review some connections between the generalized inverse and the normal equations. Originally, the training of ELMs contains two parts. The first is to randomly assign values in a specified interval to the weights between the input and hidden layers while the second is to determine the weights between the hidden and output layers by computing the generalized inverse of the matrix $\mathbf{H}$ as $\beta^* = \mathbf{H}^\dagger\mathbf{T}$. It is the minimum norm and minimum least square solution of the system of linear matrix equations $\mathbf{H}\beta = \mathbf{T}$. It is easy to prove that, if the matrix $\mathbf{H}$ is of full-rank, the solution of normal equation $\mathbf{H}^{\mathrm{T}}\mathbf{H}\beta = \mathbf{H}^{\mathrm{T}}\mathbf{T}$ is identical to $\beta^* = \mathbf{H}^\dagger\mathbf{T}$.

Noting that in Section II-A, the training process of ELMs is written as $\beta^* = (\mathbf{H}^{\mathrm{T}}\mathbf{H}+\mu\mathbf{I})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T}$, where μ is a regularizing factor. This formula is identical to $\beta^* = \mathbf{H}^\dagger\mathbf{T}$ if the regularizing factor takes value zero. It is proven in [24] that the matrix $\mathbf{H}$ is of full-rank with probability 1, and therefore, we can say that the solution of normal equation $\mathbf{H}^{\mathrm{T}}\mathbf{H}\beta = \mathbf{H}^{\mathrm{T}}\mathbf{T}$ is available with probability 1. In fact, the regularizing factor, which makes the solved weights as small as possible, has the effect to become the matrix $\mathbf{H}$ full of rank.

Practically the number of rows is much larger than the number of columns for an input data matrix. It implies that the transformation from computing $\beta^* = \mathbf{H}^\dagger\mathbf{T}$ to solving the normal system of linear matrix equations $\mathbf{H}^{\mathrm{T}}\mathbf{H}\beta = \mathbf{H}^{\mathrm{T}}\mathbf{T}$ can save much computational load, since the order of $\mathbf{H}$ is $N \times \widetilde{N}$ but the order of $\mathbf{H}^{\mathrm{T}}\mathbf{T}$ is $\widetilde{N} \times c$, where N is the number of input samples, $\widetilde{N}$ is the number of hidden layer nodes, and c is the number of classes. A lot of numerical experiments have confirmed this saving of computational load.

III. Dependency Relation Between Generalization and Uncertainty of Classifiers

In this section, we will introduce the generalization and uncertainty of a classifier. The dependency relation between generalization and uncertainty is then discussed.

A. Generalization and Uncertainty

Generally speaking, the purpose of learning is to acquire the knowledge hidden in the data. Knowledge representation, which has been well acknowledged as a bottle-neck problem in machine learning and artificial intelligence for many years, does not have a general definition but has many specific forms. A mathematical model, such as a set of IF-THEN rules or a neural network learned from a training set, can be regarded as a typical form of knowledge representation. The ability or performance of the learned model to predict unseen cases (which are not within the training set) is called generalization.

Let $\mathcal{S}$ be a finite space of samples, $F(\mathbf{x})$ be a discrete-valued function defined on $\mathcal{S}$, and $\mathbb{X}$ be a subset of $\mathcal{S}$. Based on values of $F(\mathbf{x})$ in $\mathbb{X}$, an estimator function $f(\mathbf{x})$ defined on $\mathcal{S}$ is given by using a training algorithm. The discrete-valued function $f(\mathbf{x})$ has the same value range as $F(\mathbf{x})$. Usually we call $f(\mathbf{x})$ as a classifier trained by the algorithm on $\mathbb{X}$.

Definition 1: The generalization of classifier $f(\mathbf{x})$ is defined as

$$G(f) = \frac{|\{\mathbf{x} : \mathbf{x} \in \mathcal{S} - \mathbb{X}, F(\mathbf{x}) = f(\mathbf{x})\}|}{|\mathcal{S} - \mathbb{X}|} \tag{6}$$

where $|\ |$ denotes the number of elements in a set.

Generalization is the most important index of evaluating a learned model. From mathematical viewpoint, the task of learning is to find a function $f(\mathbf{x})$ through a training set $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N \subset \mathcal{R}^n \times \{0,1\}^c$ such that $f(\mathbf{x})$ can well approximate the objective function $F(\mathbf{x})$ both at training cases and unseen cases. The difference between $F(\mathbf{x})$ and $f(\mathbf{x})$ is called generalization error, which can be measured from different angles. One method is to estimate an upper bound for it, the other is to compute $R = \int_{\mathcal{S}} [F(\mathbf{x}) - f(\mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x}$, where $p(\mathbf{x})$ is the probability density function of input $\mathbf{x}$. Experimentally, the generalization can be measured by the prediction accuracy of the classifier on a testing set.

Multiple factors have critical impacts on the generalization of a classifier.

1) *Model Selection:* It is hard to select the most appropriate model for a given classification task. When the training data is fixed, the generalizations of two models might be quite different. This is due to the data distribution, i.e., a model suitable for one type of data may not be appropriate for another type of data.
2) *Training Algorithm:* When a model is fixed, the subsequent work is to train the model parameters based on a training set. A model with a set of trained parameters has the generalization quite different from the model with another set of trained parameters.
3) *Representatives of Training Data:* Since both the objective function and its approximating function are defined on a space $\mathcal{S}$, one problem is that the training set $\mathbb{X}$ should be a reasonable sampling of the space $\mathcal{S}$, which directly relates to the fundamental assumption of machine learning that the training set has an identical distribution as the testing set has.
4) *Model Knowledge Parameters:* Different from the parameters inside the model that are acquired directly

from the training process, model knowledge parameters do not explicitly appear in the model, which are usually evaluated after the training process. For example, the uncertainty of classifier's outputs is a typical model knowledge parameter. The relationship between generalization and uncertainty of a classifier is initially demonstrated in [20]. This paper will conduct further studies on this relationship through incorporating a new index, i.e., complexity of classification.

B. Fuzziness of Classifier's Outputs

In this paper, we use fuzziness to depict the uncertainty of a classifier's outputs. The term "fuzziness," in conjunction with the concept of fuzzy set, was first mentioned by Zadeh [37]. He also generalized a probability measure of events that cannot be described by sharply defined collection of points, and suggested using entropy in information theory to interpret the uncertainty associated with a fuzzy event. De Luca and Termini [38] for the first time clearly proposed three properties that a fuzziness measure should satisfy. The term fuzziness can be interchangeable with "ambiguity" in some scenarios. Klir *et al.* [39], [40] stated that fuzziness and ambiguity gave two cognitive uncertainty measures.

As stated in [41], the fuzziness of a fuzzy set μ can be measured by a mapping $E(\mu)$:$F(\mathcal{S}) \rightarrow [0, \infty]$ where $F(\mathcal{S})$ denotes the space of all fuzzy sets defined on $\mathcal{S}$, satisfying the following axioms.

1) $E(\mu) = 0$ if and only if μ is a crisp set.
2) $E(\mu)$ attains its maximum value if and only if $\forall \mathbf{x} \in \mathcal{S}$: $\mu(\mathbf{x}) = 0.5$.
3) If $\mu \leq_s \sigma$, then $E(\mu) \geq E(\sigma)$, where $\leq_s$ is defined as

$$\mu \leq_s \sigma \Leftrightarrow \min(0.5, \mu(\mathbf{x})) \geq \min(0.5, \sigma(\mathbf{x})) \quad \max(0.5, \mu(\mathbf{x})) \leq \max(0.5, \sigma(\mathbf{x})).$$

4) $E(\mu) = E(\mu')$ when $\forall \mathbf{x} \in \mathcal{S}$: $\mu'(\mathbf{x}) = 1 - \mu(\mathbf{x})$.
5) $E(\mu \cup \sigma) + E(\mu \cap \sigma) = E(\mu) + E(\sigma)$.

Based on these axioms, we further introduce the following definition.

Definition 2 [32]: Let $B = \{\mu_1, \mu_2, \ldots, \mu_m\}$ be a fuzzy set, the fuzziness of B can be defined as

$$E(B) = -\frac{1}{m} \sum_{i=1}^{m} \big(\mu_i \log \mu_i + (1 - \mu_i) \log(1 - \mu_i)\big). \quad (7)$$

It is easy to verify that formula (7) indeed satisfies axioms 1–5.

Given a set of samples $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \subset \mathcal{R}^n \times \{0, 1\}^c$ and a well-trained classifier, a membership matrix $\mathbf{U} = [\mu_{ij}]$ can be obtained by matching each sample to the classifier, where $\mu_{ij} = \mu_j(\mathbf{x}_i)$ denotes the membership degree of the ith sample belonging to the jth class, where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, c$. It is worth noting that each output vector may not be a probability distribution, i.e., $\mu_{ij} \in [0, 1]$, and the equality $\sum_{j=1}^{c} \mu_{ij} = 1$ does not necessarily hold.

Based on Definition 2, the fuzziness of the classifier's outputs for the ith sample can be expressed as

$$E(\mu_i) = -\frac{1}{c} \sum_{j=1}^{c} \big(\mu_{ij} \log \mu_{ij} + \big(1 - \mu_{ij}\big) \log(1 - \mu_{ij})\big). \quad (8)$$

Having the above preliminaries, in the following, we propose a new concept to describe the fuzziness of a classifier's outputs on the entire training set.

Definition 3 (Fuzziness of a Classifier's Outputs): Suppose that a classifier is trained from training set $\mathbb{X}$. Without loss of generality, $\mathbb{X}$ is assumed to be a sufficient sampling of the entire sample space. Let $\mathbf{U} = [\mu_{ij}]_{c \times N}$ be the membership matrix given by matching each training sample to the classifier, where c is the number of classes and N is the number of samples. Then the fuzziness of the classifier's outputs can be defined as

$$E(\mathbf{U}) = -\frac{1}{cN} \sum_{i=1}^{N} \sum_{j=1}^{c} \big(\mu_{ij} \log \mu_{ij} + \big(1 - \mu_{ij}\big) \log\big(1 - \mu_{ij}\big)\big). \quad (9)$$

It is noted that Definition 3 uses the fuzziness of the classifier's outputs on the training set. In a more rigorous manner, it should be defined as the fuzziness of the classifier on the whole space. Unfortunately, the fuzziness of the classifier on unseen samples is unknown. According to the fundamental assumption of supervised learning that the training set is a reasonable and sufficient sampling of the entire sample space, we can use the classifier's fuzziness on the training set to approximately replace the classifier's fuzziness on the entire sample space.

C. Relationship Between Generalization and Fuzziness

Previous study [20] shows that the classifier with higher fuzziness of outputs has a better generalization for complex boundary problems when the training accuracy attains a predefined threshold. Furthermore, it demonstrates that the outputs of boundary samples have higher fuzziness, and samples with higher fuzziness exhibit higher risk of misclassification. By separating samples with high fuzziness from samples with low fuzziness, a divide-and-conquer learning algorithm based on fuzziness categorization was proposed in [41]. It shows that the category of sample with low or high fuzziness plays a critical role for performance improvement. Although these studies confirm that a relationship between fuzziness and generalization of a classifier indeed exists, it is difficult to explicitly express this relationship in general.

In the following, we make an investigation on data set *Spam*, which is a binary classification data set selected from UCI machine learning repository. This data set contains 4601 samples with 57 features. We randomly split it into two parts, i.e., 70% for training and 30% for testing. ELM is used to construct a classifier, which generates four indexes, i.e., training accuracy, testing accuracy, training fuzziness, and testing fuzziness. The random splitting is repeated for 100 times and four indexes are recorded for each repetition.

We make a statistical analysis for the 100 results. First, we split the interval between the minimum and maximum fuzziness values into ten parts with equal length and generate ten

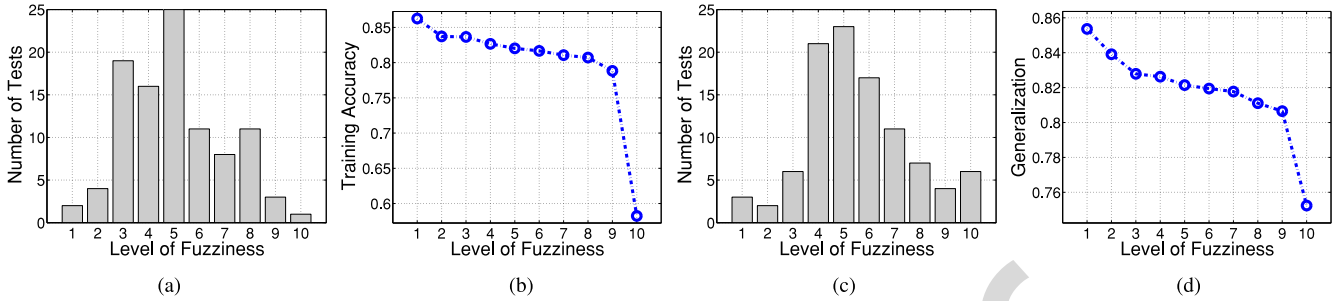


Fig. 1. Dependency relation between fuzziness and accuracy for *Spam*. (a) Histogram of training fuzziness. (b) Training accuracy. (c) Histogram of testing fuzziness. (d) Testing accuracy.

levels of fuzziness. For instance, the minimum and maximum fuzziness values for testing are 0.4889 and 0.5798, respectively. Then, the ten fuzziness levels for testing are generated as level 1 = [0.4889, 0.4980), level 2 = [0.4980, 0.5071), level 3 = [0.5071, 0.5162),..., and level 10 = [0.5707,0.5798]. Afterwards, we make a statistic for the number of experimental trials in each fuzziness level, and plot the histograms as shown in Fig. 1(a) and (c). Finally, we get the average training or testing accuracy for each fuzziness level, and plot the changing trends as shown in Fig. 1(b) and (d).

One can see from Fig. 1 that the relationship between accuracy and fuzziness of ELM does exist for *Spam*. We further calculate the Pearson correlation coefficient. As a remark, Pearson correlation reflects the statistical relationship between two sets of variables with a coefficient from $[-1, 1]$. A positive/negative coefficient represents that the two sets of variables are positive/negative correlated, and the absolute value represents the correlation degree. We use the median to represent each fuzziness level. Taking the testing result as an example, the correlation coefficient is calculated between fuzziness vector [0.4935, 0.5025, 0.5116, 0.5207, 0.5298, 0.5389, 0.5480, 0.5571, 0.5662, 0.5753] and accuracy vector [0.8536, 0.8391, 0.8279, 0.8263, 0.8214, 0.8194, 0.8177, 0.8111, 0.8065, 0.7524]. Finally, the correlation coefficients for training and testing are calculated as -0.7145 and -0.8625, respectively. This tells that the accuracy and fuzziness have a negative correlation for *Spam*, i.e., a higher fuzziness will lead to a lower accuracy, and the correlation degree is high.

Although the above example demonstrates that the relationship between generalization and uncertainty does exist for data set *Spam*, this relationship is difficult to express explicitly for general cases. In the subsequent sections, we will attempt to make this relationship clear by incorporating a new index, i.e., complexity of classification.

IV. Complexity of Classification Problem

Generally, a classification problem can be described as follows. Let $\mathcal{S}$ be the universal space we consider, F be a discrete function defined on $\mathcal{S}$. For simplicity, we suppose that function F takes values either 0 or 1, where 0 denotes one class and 1 denotes the other class. Given a subset of $\mathcal{S}$, denoted as $\mathbb{X}$, which is called the training set, the values of F on $\mathbb{X}$ are known, but the values of F on $\mathcal{S} - \mathbb{X}$ are unknown. A classification problem is to find a function f such that f can well approximate F both in $\mathbb{X}$ and $\mathcal{S} - \mathbb{X}$. Usually, F is called an objective function, f is called a classifier acquired based on training set $\mathbb{X}$, the approximation error on $\mathbb{X}$ is called training error, and the approximation error on $\mathcal{S} - \mathbb{X}$ represents the generalization ability of F.

The complexity of a classification problem refers to the complexity of function F, which implies the difficulties of the process of finding a quality f from $\mathbb{X}$. Unfortunately, there is no formal definition on the complexity of a discrete function. From references we can find a number of indexes to describe the complexity from different angles. It is noteworthy that the complexity of objective function is independent on the learned classifier f. Since the objective function F is unknown in real applications but is known on the training set $\mathbb{X}$, the indexes in describing the complexity of F can be estimated through the training set $\mathbb{X}$ and values of F on $\mathbb{X}$. In the following, we give several indexes to describe the complexity of F, which are mainly chosen from [21].

A. Fisher's Discriminant Ratio

Fisher's discriminant ratio is an old statistical index for describing the difference between two populations. Suppose that μ_{1j}, μ_{2j}, σ_{1j}, and σ_{2j} are the means and variances of the two populations (classes) with respect to the jth attribute, $j = 1, \ldots, n$. Then, the Fisher's discriminant ratio for the jth attributes is defined as

$$\mathfrak{f}_j = \frac{\left(\mu_{1j} - \mu_{2j}\right)^2}{\sigma_{1j}^2 + \sigma_{2j}^2}. \tag{10}$$

It is easy to see that Fisher's discriminant ratio with respect to the jth attribute describes the distance between two classes regarding this attribute. Intuitively, the longer the distance is, the easier the classification problem is, the lower the complexity will be. Thus, the complexity evaluating index is defined as

$$\mathfrak{Comp}_1 = \frac{1}{\max_j\{\mathfrak{f}_j\}}. \tag{11}$$

B. Volume of Overlap Region

A similar measure is the volume of overlap region between two class conditional distributions. It depends on, for each attribute, the maximum and the minimum values of each class.

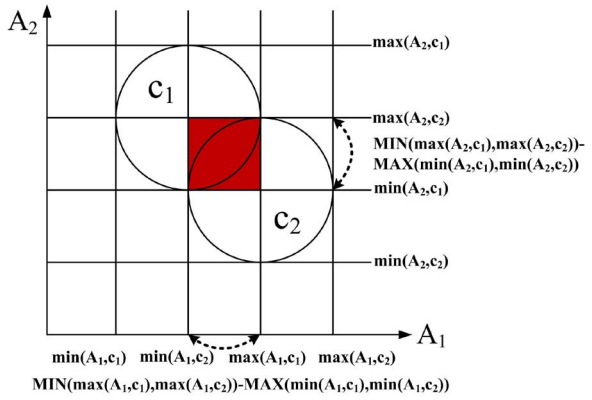


Fig. 2. Intuitive illustration of volume of overlap region.

We denote A_j as the jth attribute. Then, the overlap region normalized by the range of the value spanned by both classes, for each attribute A_j, can be represented as

$$\mathfrak{v}_j = \frac{\mathrm{MIN}\big(\max(A_j, c_1), \max(A_j, c_2)\big) - \mathrm{MAX}\big(\min(A_j, c_1), \min(A_j, c_2)\big)}{\mathrm{MAX}\big(\max(A_j, c_1), \max(A_j, c_2)\big) - \mathrm{MIN}\big(\min(A_j, c_1), \min(A_j, c_2)\big)} \tag{12}$$

where $\max(A_j, c_1)$, $\max(A_j, c_2)$, $\min(A_j, c_1)$, and $\min(A_j, c_2)$ denotes the maximum and minimum values of attribute A_j in the two classes, respectively. Then, the complexity evaluating index is defined as the volume of overlap region incorporating all the attributes

$$\mathfrak{Comp}_2 = \prod_{j=1}^{n} \mathfrak{v}_j \tag{13}$$

where n is the number of attributes. An intuitive illustration of volume of overlap region for a 2-D feature space is given in Fig. 2. It is noted that $\mathfrak{Comp}_2 = 0$ if the value ranges of the two classes do not overlap in at least one dimension. Obviously, a larger value of $\mathfrak{Comp}_2$ represents a higher complexity of the classification problem.

C. Intraclass/Interclass Distance Ratio

This measure first computes the Euclidean distance from each sample to its nearest neighbor within or outside the class. Assume that d_i^{intra} or d_i^{inter} is the distance between sample $\mathbf{x}_i$ and its nearest neighbor within or outside the class, we have

$$\begin{cases} d_i^{\mathrm{intra}} = \min_{j \neq i, y_j = y_i} d(\mathbf{x}_i, \mathbf{x}_j) \\ d_i^{\mathrm{inter}} = \min_{j \neq i, y_j \neq y_i} d(\mathbf{x}_i, \mathbf{x}_j) \end{cases} \tag{14}$$

where y_i and y_j represent the class labels of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Then, it takes the average of all the intraclass distances and the average of all the interclass distances, and the ratio of both averages is defined as the complexity of the problem

$$\mathfrak{Comp}_3 = \frac{\sum_{i=1}^{N} d_i^{\mathrm{intra}}}{\sum_{i=1}^{N} d_i^{\mathrm{inter}}} \tag{15}$$

where N is the number of samples. Similarly, a larger value of $\mathfrak{Comp}_3$ represents a higher complexity of the classification problem.

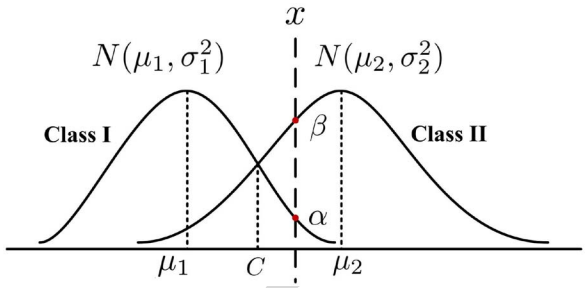


Fig. 3. Two normal populations.

D. Linear Separability

Linear separability was intensively discussed in the early literature. A simple definition to describe the linear separability for both separable and nonseparable cases is given by Smith [42]

$$\min \ \mathbf{a}^{\mathrm{T}}\mathbf{t}, \quad \text{s.t. } \mathbf{Z}^{\mathrm{T}}\mathbf{w} = \mathbf{t} \geq \mathbf{b} \tag{16}$$

where $\mathbf{a}$ and $\mathbf{b}$ are arbitrary constant vectors, $\mathbf{w}$ is the weight vector, $\mathbf{t} \geq 0$ is the error vector, and $\mathbf{Z}$ is a matrix in which each column $\mathbf{z}$ is defined based on the input vector $\mathbf{x}$ and its class label $\mathfrak{c}$

$$\begin{cases} \mathbf{z} = +\mathbf{x} & \text{if } \mathfrak{c} = c_1 \\ \mathbf{z} = -\mathbf{x} & \text{if } \mathfrak{c} = c_2. \end{cases} \tag{17}$$

The value of the objective function denotes the degree of being separable for two class cases, that is

$$\mathfrak{Comp}_4 = \mathbf{a}^{\mathrm{T}}\mathbf{t}. \tag{18}$$

It is noted that $\mathfrak{Comp}_4 = 0$ if the problem is linear separable.

Other indexes to describe the complexity of classification problem can be found from [21].

V. Relationship Between Generalization and Uncertainty by Incorporating Complexity of Classification

In this section, we give an analysis on the relationship between generalization and uncertainty by incorporating the complexity of classification. Since it is difficult for us to give a general analysis for all the complexity indexes, we only adopt the index of Fisher's discriminant ratio in Section IV-A, and give an explanation from the viewpoint of discriminant analysis, which has the principal of maximum probability.

Without loss of generality, we consider the 1-D case, which can be easily extended to multiple-dimensional cases. A normal distribution with mean μ and variance σ^2, denoted by $N(\mu, \sigma^2)$, has a probability density function

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \ -\infty < x < +\infty. \tag{19}$$

Suppose that there are two normal populations denoted by $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ as shown in Fig. 3, and $x(\mu_1 < x < \mu_2)$ is a new sample that needs to be discriminated.

For a classification problem, each population represents a class. From traditional textbook [43] we can view a simple way to judge sample x belonging to which class.

Let C be the cross-point between two density functions, i.e., C satisfies the following equation:

$$\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{(C-\mu_1)^2}{2\sigma_1^2}\right)=\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{(C-\mu_2)^2}{2\sigma_2^2}\right). \tag{20}$$

It is easy to check that the cross-point locates in the interval (μ_1, μ_2). The probabilities of sample x belonging to the two classes, denoted as (α, β), can be approximately viewed as

$$(\alpha,\beta)=\left(\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right),\ \frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)\right) \tag{21}$$

which induces the following discriminant rules based on the principle of maximum probability.

1) IF $x < C$ ($\alpha > \beta$) THEN x belongs to class I.
2) IF $x > C$ ($\alpha < \beta$) THEN x belongs to class II.
3) IF $x = C$ ($\alpha = \beta$) THEN the class of x is uncertain.

We now relate these discussions about discriminant analysis to the theme of this paper, i.e., uncertainty and complexity of a classification problem. According to Section IV-A, the complexity of a classification problem can be described by means and variances of class distributions. It can be roughly summarized as: the complexity is going up with either increasing the variances (σ_1^2, σ_2^2) or decreasing the difference between both means $|\mu_1 - \mu_2|$. Moreover, the uncertainty of a classifier is evaluated based on the probability vector (α, β) defined in (21). According to Section III, there are many specific formulas to evaluate the uncertainty (e.g., the fuzziness in Definition 3), but all of them have to satisfy the conditions given in Section III-B, e.g., if $\alpha < \beta$, when $\alpha' < \alpha$ and $\beta' > \beta$, the uncertainty output by vector (α', β') should be smaller than that output by (α, β). It shows that, to some extent, the difference between the two probability values denotes the magnitude of uncertainty. The bigger the difference is, the smaller the uncertainty is. Based on these analyses, we have the following theorems.

Theorem 1: Let

$$g(\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\left(\exp\left(-\frac{(x-\mu_2)^2}{2\sigma^2}\right)-\exp\left(-\frac{(x-\mu_1)^2}{2\sigma^2}\right)\right)$$

where $\sigma > 0$, $\mu_1 < \mu_2$, $x \in ([(\mu_1 + \mu_2)/2], \mu_2)$, and μ_1, μ_2 are considered as constants. Then, there exists a number $\sigma_1 \in (0, \mu_2 - x)$ such that $g(\sigma)$ is monotonically decreasing in the interval $(\sigma_1, +\infty)$.

Proof: The proof of Theorem 1 is listed in Appendix B. ■

Theorem 2: Let

$$q(\delta)=\frac{1}{\sqrt{2\pi}}\left(\exp\left(-\frac{x-(\mu_2-\delta)^2}{2}\right)-\exp\left(-\frac{x-(\mu_1+\delta^*)^2}{2}\right)\right)$$

where x, μ_1, and μ_2 are considered as constants, $\mu_1 < \mu_2$, $\delta^* = |[(\mu_1 - x)/(\mu_2 - x)]|\delta$, and $\delta > 0$. Then, there exists a number δ_1 such that $q(\delta)$ is monotonically decreasing in the interval $(0, \delta_1)$.

Proof: The proof of Theorem 2 can be derived similarly to the proof of Theorem 1. ■

Theorem 3: Suppose that the conditional probability outputs of a binary classifier follow two normal distributions $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$, respectively, where $\mu_1 < \mu_2$. Let

$$\alpha=-\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu_1)^2}{2\sigma^2}\right)\beta=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma^2}\right)$$

and

$$E(\alpha,\beta)=-\frac{1}{2}(\alpha\log\alpha+(1-\alpha)\log(1-\alpha)+\beta\log\beta+(1-\beta)\log(1-\beta)).$$

Assume $\beta = K\alpha$ where $K \in (1, 1+\epsilon)$, then $E(\alpha, \beta) = E(K)$ is monotonically decreasing with respect to K if $K\alpha > (1/2)$.

Proof: The proof of Theorem 3 is listed in Appendix C. ■

Noting that $g(\sigma)$ in Theorem 1 or $q(\delta)$ in Theorem 2 denotes the difference between two probability density values, which can be represented as $\beta - \alpha$ in Theorem 3. Theorem 3 directly connects this difference together with the uncertainty of the classifier's outputs given in Definition 2.

Theorem 3 shows that the uncertainty of the classifier's outputs is decreasing with the increase of the difference between two density values, i.e., $\beta - \alpha$, where α and β can be considered as the probabilities of a sample being classified as classes I and II, respectively. As a result, the conclusions in Theorems 1 and 2 show that the uncertainty of a classifier's outputs is becoming bigger with the increase of the complexity of the classification problem, which is represented through inflating the variance in Theorem 1 and through shrinking the difference between two means in Theorem 2, respectively. Since in a classification problem, the complexity is inherent while the uncertainty is generated by the output of a well-trained classifier which has its training and testing accuracy, it is reasonable to believe that some relationships exist among the accuracy, uncertainty, and complexity.

It is noteworthy that Theorems 1–3 cannot exactly explain the relationships among the three indexes, i.e., accuracy, uncertainty, and complexity. However, to a great extent, they provide solid supports to the existence of the relationships. They confirm such a fact that the classifier's uncertainty will be inevitably high if the classification problem is complex, no matter what classifier design algorithm is used. This statement further implies that a high-performance classifier will have high uncertainty when the problem is complex.

VI. Empirical Studies

In this section, we will conduct some empirical studies to further analyze the relationships discussed in Section V. It is noteworthy the discussions in Section V were made based

TABLE I
SELECTED DATA SETS FOR EXPERIMENTS

No	Data Set	# Samples	# Features	# Classes
1	Libras	369	90	15
2	Breast	699	9	2
3	SPECTF	267	44	2
4	Cancer	683	9	2
5	Chart	600	60	6
6	Cotton	356	21	6
7	CT	221	36	2
8	Dermatology	366	34	6
9	Ecoli	336	7	8
10	German	1,000	24	2
11	Glass	214	9	6
12	Haberman	306	3	2
13	Heart	270	13	2
14	Vowel	990	10	11
15	Ionosphere	351	34	2
16	Australian	690	14	2
17	Pima	768	8	2
18	Plrx	182	12	2
19	Sonar	208	60	2
20	Soybean	683	35	19
21	Bupa	345	6	2
22	Transfusion	748	4	2
23	Segment	2,310	19	7
24	Wdbc	569	30	2
25	Wpbc	198	33	2
26	Yeast	1,484	8	10
27	Zoo	101	16	7
28	Spam	4,601	57	2
29	Satellite	6,435	36	6
30	OptDigits	5,620	64	10
31	Pen	10,992	16	10

on $\mathfrak{Comp}_1$, i.e., Fisher's discriminant ratio. Thus, in this section, we will also adopt $\mathfrak{Comp}_1$ to evaluate the complexity of classification problems.

A. Selected Data Sets

The data sets used in the experiments are selected from UCI machine learning repository. The detailed information regarding these data sets are summarized in Table I. Since the complexity indexes listed in Section IV are defined for binary classification problems, we transfer each multiclass data set into binary by randomly selecting 50% classes as positive and the rest 50% classes as negative.

B. Experimental Design

The flowchart for training the classifier and evaluating the problem complexity is listed in Algorithm 1.

It is noteworthy that the training algorithm adopted in this section is ELM. Due to the random mechanism for weight assignment, it is easy to repeat the experiment for many times. We conduct 100 experimental trials for each data set. In each trial, 70% data are randomly selected for training, and the remaining 30% data are used for testing. Each trial will provide a different result, and we make statistics for fuzziness, accuracy, and complexity based on the 100 results.

The number of hidden nodes in ELM is set as 20, and sigmoid activation function is utilized. The simulations are carried out under MATLAB R2011b, which are executed on a computer with an Intel Core i7-5500U CPU@2.40 GHz, 8GB memory, and 64-bit Windows 8 system.

Algorithm 1: Train ELM Classifier and Compute Evaluating Indexes

Input:
Training set $\mathbb{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \subset \mathcal{R}^n \times \{0, 1\}^c$;
Activation function $f(\mathbf{x})$;
Number of hidden nodes $\tilde{N}$.

Output:
Fuzziness and generalization of the trained classifier;
Complexity of the classification problem.

1 *Data processing*: randomly divide the data set into two parts for training and testing according to a separation ratio.
2 *Classifier training*: train a ELM classifier based on the algorithm given in section II-A.
3 *Testing*: test the classifier on the testing set, compute the fuzziness (Definition 3) and generalization (testing accuracy) of the classifier.
4 *Complexity evaluation*: compute the complexity of the classification problem, i.e., Eq. (11).

C. Experimental Analysis

Similar to Section III-C, we make some statistical analyses on the testing results. For each data set, ten fuzziness levels are generated by equally dividing the interval between the maximum and minimum fuzziness values. We use the median to represent each fuzziness level. Then, the number of experimental trials for each fuzziness level is counted, and the average testing accuracy for each fuzziness level is calculated. Fig. 4 demonstrates the changing trend of the testing accuracy along with the level of fuzziness. It depicts the dependency relation between testing accuracy and testing fuzziness for the classification problems. Due to space limit, we only plot the results for 12 data sets out of 31. Furthermore, we calculate the Pearson correlation coefficient between fuzziness vector and accuracy vector for each data set. It is noteworthy there are ten fuzziness levels for each data set. However, from Fig. 4, we can see that the highest fuzziness level (i.e., level ten) usually cause a sharp change of the testing accuracy, which may interfere the statistical analysis for the overall results. Thus, we only use the previous nine fuzziness values and their corresponding accuracy. The correlation coefficients r are listed in Table II. We artificially set up some thresholds to justify the degree of correlation.

1) If $0 \leq |r| < 0.4$, then the correlation is low.
2) If $0.4 \leq |r| < 0.7$, then the correlation is medium.
3) If $0.7 \leq |r| \leq 1$, then the correlation is strong.

It is observed from Table II that the generalization and fuzziness have a strong or medium correlation regarding most data sets.

The complexities of the problems are shown in Fig. 5, which are sorted according to the order numbers (i.e., 1–31) in Table I. In Fig. 5, we artificially set up a threshold such that the complexity higher than the threshold is called high otherwise is called low. In this case, one can view an implicit relation among the complexity, generalization, and fuzziness.

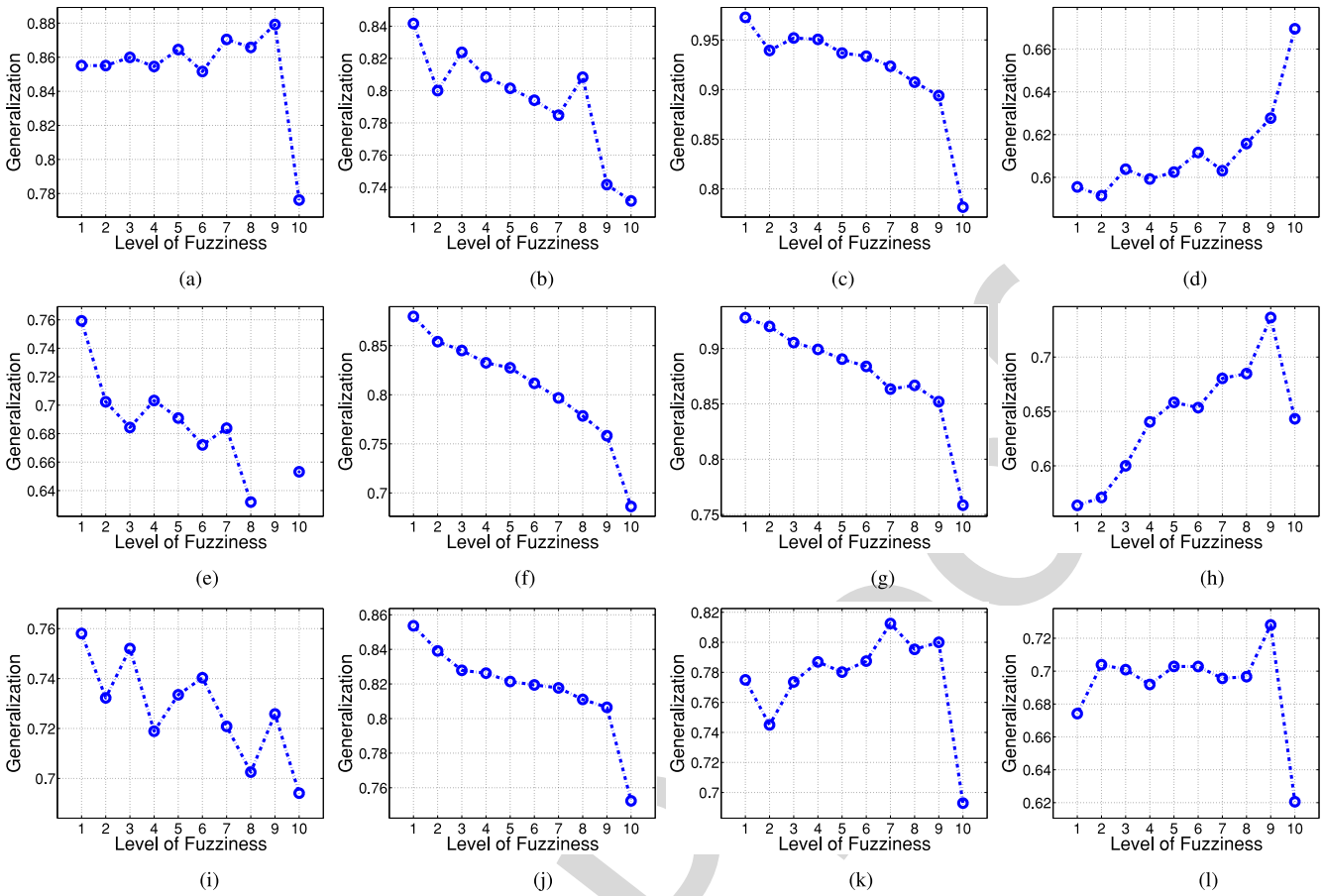


Fig. 4. Relationship between fuzziness and generalization of ELM classifier on different data sets. (a) Australian. (b) Chart. (c) Dermatology. (d) Segment. (e) Libras. (f) OptDigits. (g) Pen. (h) Plrx. (i) Sonar. (j) Spam. (k) SPECTF. (l) Yeast.

TABLE II
PEARSON CORRELATION COEFFICIENT BETWEEN OUTPUT FUZZINESS AND TESTING ACCURACY

Data Set	Pearson Correlation Coefficient	Data Set	Pearson Correlation Coefficient
1	-0.6434√	17	-0.0520†
2	-0.3522†	18	0.9743★
3	0.7838★	19	-0.6896√
4	-0.7835★	20	-0.9348★
5	-0.7718★	21	0.4132√
6	0.3421†	22	0.4962√
7	0.3744†	23	0.8728★
8	-0.9277★	24	-0.2933†
9	-0.0579†	25	-0.5559√
10	-0.1474†	26	0.6297√
11	-0.5452√	27	0.3470†
12	0.5803√	28	-0.9496★
13	0.1768†	29	0.1455†
14	-0.9362★	30	-0.9903★
15	0.5782√	31	-0.9895★
16	0.7420★		

Note: For each data set, ★ represents strong correlation, √ represents medium correlation, and † represents low correlation.

The generalization of a classifier trained by ELM goes up with the increase of fuzziness if the complexity of the classification problem is relatively high, while the generalization of a classifier trained by ELM goes down with the increase of fuzziness if the complexity of the classification problem is relatively low. For instance, it can be seen from Fig. 5 that the complexity values of *Segment* (data set 23) and *Plrx* (data set 18) are high, in this case, the generalizations of these two data sets are becoming better with the increase of fuzziness as shown in Fig. 4(d) and (h). However, the complexity values of *OptDigits* (data set 30) and *Spam* (data set 28) are low, in this case, the generalizations of these two data sets are becoming worse with the increase of fuzziness as shown in Fig. 4(f) and (j).

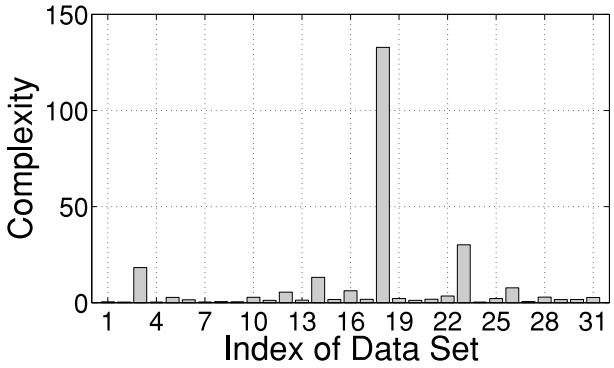


Fig. 5. Complexity of the classification problems.

By learning the complexity of classification problems from Fig. 5, we grasp some factors that are resulted from the

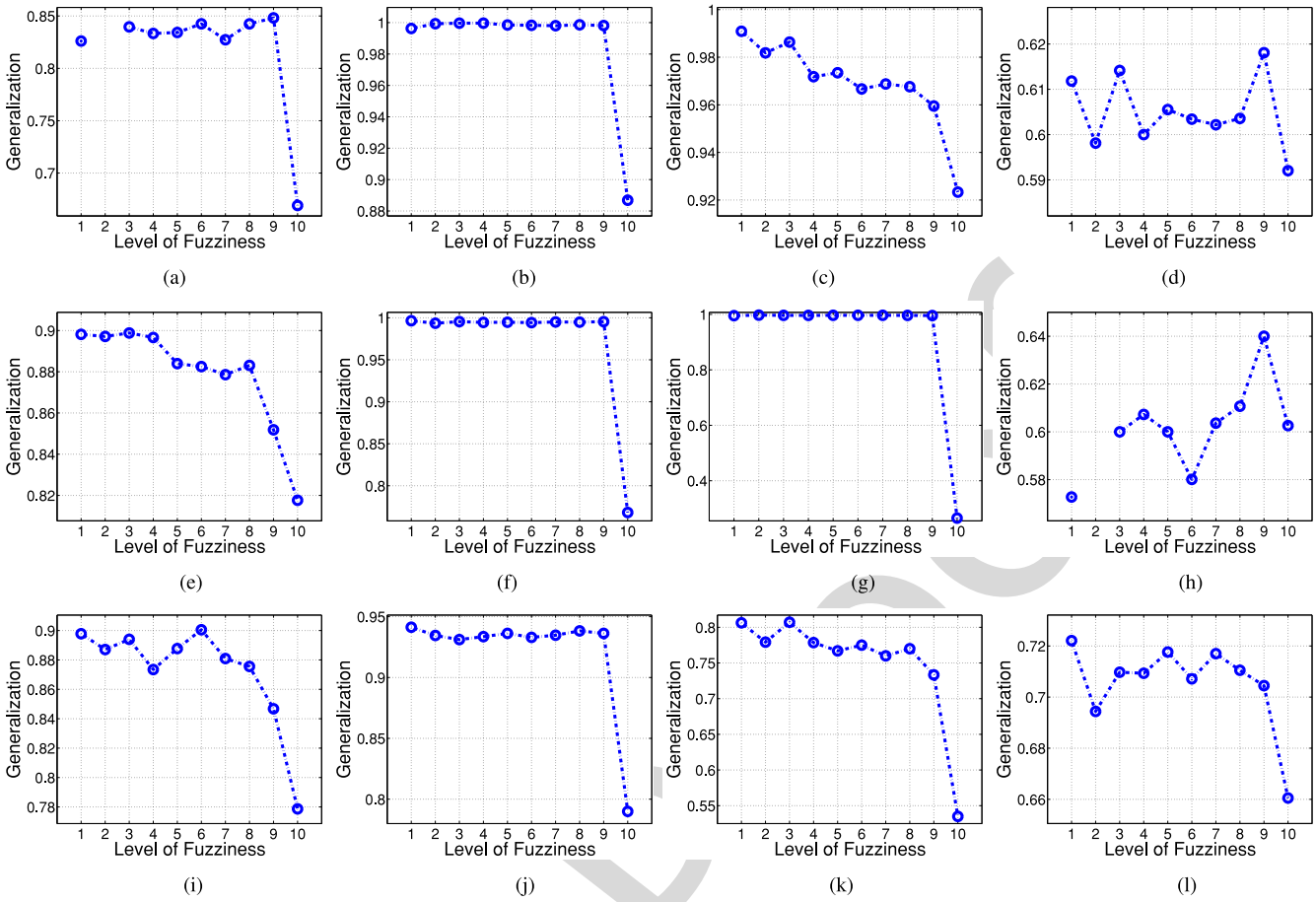


Fig. 6. Relationship between fuzziness and generalization of SVM classifier on different data sets. (a) Australian. (b) Chart. (c) Dermatology. (d) Segment. (e) Libras. (f) OptDigits. (g) Pen. (h) Plrx. (i) Sonar. (j) Spam. (k) SPECTF. (l) Yeast.

complexity of decision boundaries. It is obvious that there are some relations between them.

As we know, the complexity of a classification problem can be intuitively regarded as the degree of difficulty for the problem. More specifically, it is the complexity of geometrical class boundary which can be seen as an equation $F = 0$ that divides the sample space. In classification problem, it is desired to find a classifier f by training the data set locating next to the boundary function $F = 0$. The ability of function f to approximate function F on unseen data is the generalization, and the fuzziness of the classifier is the uncertainty of function f in dividing unseen samples.

When it is easy to distinguish the classes by the boundary of function F, it will also be easy to divide the unseen samples by f, since the structure of training data is supposed to be similar to the structure of unseen data and f is an estimator of F. It implies that the boundary will be simple and the fuzziness of the boundary is low. In this situation, it is reasonable to believe that, with the decrease of classifier's fuzziness, the generalization will be improved.

When it is difficult to distinguish the classes by the boundary of function F, the classifier function f is also difficult to divide the unseen samples. It corresponds to a case of high complexity and complex boundary. It is inherent to output high fuzziness for boundary samples for any classifier, and therefore, we reasonably believe in this situation that, with the increase of classifier's fuzziness, the generalization may be getting better.

D. Analysis With SVM Classifiers

We further realize the above studies with SVM classifiers. We adopt the "LibSVM" toolbox, the penalty term C is fixed as 100, and RBF kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}_i) = \exp(-[||\mathbf{x} - \mathbf{x}_i||^2/2\sigma^2])$ with $\sigma = 1$ is adopted. The decision values of SVM are transformed into uncertain outputs by logistic function. The dependency relation between generalization and fuzziness regarding the 12 data sets in Fig. 4 are demonstrated in Fig. 6. It can be observed that the results are basically consistent with those in Section VI-C, but the changing trends are not as clear as those of ELM. As a result, ELM might be more suitable to conduct this paper, since it has a higher degree of uncertainty due to the random mechanism for input weights assignment.

VII. Conclusion

This paper finds an empirical relationship among the complexity of a classification problem, the uncertainty of classifier's outputs, and the prediction accuracy of the classifier. By experimental validation and theoretical explanation through a simple model of discriminant analysis, it is found that with the

increase of the uncertainty of the classifier's outputs, empirically the accuracy is upgrading for high-complexity problem but downgrading for low-complexity problem. Based on these findings, in order to choose a better classification rule for a practical problem, one can tune the model parameters such that the uncertainty becomes larger for problems with higher complexity, or smaller for problems with lower complexity under the condition that an acceptable training accuracy is kept.

APPENDIX A
FEATURES OF ELMS

In the following, we briefly review the major advantages of ELMs.

1) The first advantage of ELMs is the fast training speed. Since the training of ELMs does not include iterative tuning, it statistically shows that ELM is thousands of times faster than BP given a predefined threshold for training accuracy.
2) Another feature of ELMs is the acceptable generalization ability. In comparison with other popular classification or regression algorithms, such as DTs, SVMs, logistic regressions, etc., the generalization of ELMs may not be the best in general. But so far, one cannot find a significant difference among the generalizations of these algorithms.
3) The training procedure of ELMs can process online sequential data conveniently, which demonstrates strong potentials for big data analytic. It is shown that ELMs can effectively handle both numerical and nominal attributes for both classification and regression problems.
4) Mathematically it is proven that ELMs have the universal approximation ability if the activation function is differentiable. That is, ELMs can uniformly approximate any continuous function defined in an interval when the number of hidden nodes goes to infinity. This conclusion establishes the foundation of applying ELMs to various classification and regression problems.

It is worthy noting that any learning algorithm cannot be consistently better than others. In the following, we list several disadvantages of ELMs.

1) As aforementioned, the weights between input and hidden layers in ELMs are randomly selected from an interval. ELMs are sensitive to this interval, and the change of the interval will produce quite different classifiers, which seriously decreases the stability.
2) The number of hidden layer nodes is critical for building an ELM. A large number will lead to the generalization decreasing but a small number can result in the training error increasing. So far, how to select the number of hidden layer nodes is still a challenging issue.

APPENDIX B
PROOF OF THEOREM 1

The original problem can be represented as

$$g(\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\left(\exp\left(-\frac{(x-b)^2}{2\sigma^2}\right) - \exp\left(-\frac{(x-a)^2}{2\sigma^2}\right)\right)$$

prove that there exits σ_1 such that $g(\sigma)$ is monotonically increasing when $\sigma < \sigma_1$ and $g(\sigma)$ is monotonically decreasing when $\sigma > \sigma_1$.

The constant term $\sqrt{2\pi}$ can be neglected. Let $(x-a) = k \times (b-x)$ and $\sigma = t \times (b-x)$, the original problem can be simplified as

$$g(t) = \frac{1}{t}\left(\exp(-\frac{1}{2t^2}) - \exp(-\frac{k^2}{2t^2})\right), \quad k > 1 \text{ and } t > 0$$

prove that there exits t_1 such that $g(t)$ is monotonically increasing when $t < t_1$ and $g(t)$ is monotonically decreasing when $t > t_1$.

We get the first-order derivation of $g(t)$, that is

$$g'(t) = \frac{1}{t^4}\left[\left(1-t^2\right)\exp\left(-\frac{1}{2t^2}\right) - \left(k^2-t^2\right)\exp\left(-\frac{k^2}{2t^2}\right)\right].$$

Having this derivation, it can be derived as follows.

1) When $t > k$, $t^2 - 1 > t^2 - k^2 > 0$ and $\exp(-[1/2t^2]) > \exp(-[k^2/2t^2])$, thus $(t^2-1)\exp(-[1/2t^2]) > (t^2-k^2)\exp(-[k^2/2t^2])$, thus we have $g'(t) < 0$.
2) When $k \geq t > 1$, $(1-t^2)\exp(-[1/2t^2]) < 0$, thus $(k^2-t^2)\exp(-[k^2/2t^2]) > 0$, thus we have $g'(t) < 0$.
3) When $t = 1$, we have $g'(t) = [1/t^4][-(k^2-t^2)\exp(-[k^2/2t^2])] < 0$.

So far, we have proved that $g'(t) < 0$ when $t \geq 1$, which means that $g(t)$ is monotonically decreasing when $t \geq 1$.

When $1 > t > 0$ and $t \to 0$, we have $[(1-t^2)/(k^2-t^2)] \to (1/k^2)$ and $\exp([(1-k^2)/2t^2]) \to 0$ (noting that $t \leq 1 < k$). There exists $t^* \in (0,1)$ such that $([(1-t^{*2})/(k^2-t^{*2})] > \exp([(1-k^2)/2t^{*2}]) = [\exp(1/2t^{*2})/\exp([k^2/2t^{*2}])]$, thus $([(1-t^{*2})\exp(-1/2t^{*2})]/[(k^2-t^{*2})\exp(-k^2/2t^{*2})]) > 1$, thus $(1-t^{*2})\exp(-1/2t^{*2}) > (k^2-t^{*2})\exp(-k^2/2t^{*2})$, thus $g'(t^*) > 0$.

According to Zero theorem, there exits $t_1 \in (0,1)$ such that $g'(t_1) = 0$. Since $g'(t)$ is continuous and differentiable, if all the stagnation points are maximum points, then there is only one stagnation point, otherwise minimum point exists.

We further get the second-order derivation of $g(t)$, that is

$$g''(t) = \frac{1}{t^7}\left\{\left[2t^2\left(t^2-1\right) - 2t^2 + (1-t^2)\right]\exp\left(-\frac{1}{2t^2}\right) - \left[2t^2\left(t^2-k^2\right) - 2t^2k^2 + k^2\left(k^2-t^2\right)\right]\exp\left(-\frac{k^2}{2t^2}\right)\right\}.$$

Put the stagnation point t_1 into $g''(t)$, since $(1-t_1^2)\exp(-1/2t_1^2) - (k^2-t_1^2)\exp(-k^2/2t_1^2) = 0$, we have

$$g''(t_1) = \frac{1}{t_1^7}\left\{-2t_1^2\left[\exp\left(-\frac{1}{2t_1^2}\right) - k^2\exp\left(-\frac{k^2}{2t_1^2}\right)\right] + \left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - k^2\left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right)\right\}.$$

Based on

$$\left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - \left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right) = 0$$

$$k > 1 \text{ and } 1 > t_1 > 0$$

we have

$$\exp\left(-\frac{1}{2t_1^2}\right) - k^2\exp\left(-\frac{k^2}{2t_1^2}\right) = t_1^2\left[\exp\left(-\frac{1}{2t_1^2}\right) - \exp\left(-\frac{k^2}{2t_1^2}\right)\right] > 0$$

and

$$\left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - k^2\left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right) < \left(1-t_1^2\right)\exp\left(-\frac{1}{2t_1^2}\right) - \left(k^2-t_1^2\right)\exp\left(-\frac{k^2}{2t_1^2}\right) = 0.$$

Thus, $g''(t_1) < 0$, t_1 is the maximum point, which means that $g(t)$ is monotonically increasing when $t < t_1$ and $g(t)$ is monotonically decreasing when $t_1 < t < 1$.

To this end, we have proved that $g(t)$ is monotonically increasing when $t < t_1$ and $g(t)$ is monotonically decreasing when $t > t_1$.

APPENDIX C
PROOF OF THEOREM 3

Substituting β with $K\alpha$ in $E(K)$, we have

$$E(K) = -\frac{1}{2}\big(\alpha\log\alpha + (1-\alpha)\log(1-\alpha) + K\alpha\log(K\alpha) + (1-K\alpha)\log(1-K\alpha)\big).$$

Taking derivative of $E(K)$ with respect to K, we obtain

$$\frac{dE(K)}{d(K)} = -\frac{1}{2}(\alpha\log(K\alpha) - \alpha\log(1-K\alpha)) = -\frac{\alpha}{2}\log\frac{K\alpha}{1-K\alpha}.$$

It is easy to view that $[dE(K)/d(K)] < 0$ if $K\alpha > (1/2)$, which completes the proof.

REFERENCES

[1] W. W. Y. Ng, A. P. F. Chan, D. S. Yeung, and E. C. C. Tsang, "Quantitative study on the generalization error of multiple classifier systems," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2005, pp. 889–894.

[2] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[3] X. D. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[4] Z. Yan and C. Xu, "Studies on classification models using decision boundaries," in *Proc. 8th IEEE Int. Conf. Cogn. Informat.*, Hong Kong, 2009, pp. 287–296.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2012.

[6] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 2000.

[7] R. Wang, S. Kwong, and D. Chen, "Inconsistency-based active learning for support vector machines," *Pattern Recognit.*, vol. 45, no. 10, pp. 3751–3767, 2012.

[8] R. Wang and S. Kwong, "Active learning with multi-criteria decision making systems," *Pattern Recognit.*, vol. 47, no. 9, pp. 3106–3119, 2014.

[9] R. Wang, D. Chen, and S. Kwong, "Fuzzy-rough-set-based active learning," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1699–1704, Dec. 2014.

[10] R. Wang, C.-Y. Chow, and S. Kwong, "Ambiguity-based multiclass active learning," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 242–248, Feb. 2016.

[11] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[12] R. Wang, S. Kwong, X.-Z. Wang, and Q. Jiang, "Segment based decision tree induction with continuous valued attributes," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1262–1275, Jul. 2015.

[13] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[14] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[15] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989.

[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[17] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[18] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 1, pp. 1–14, Feb. 1998.

[19] Z. Deng, K.-S. Choi, Y. Jiang, and S. Wang, "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2585–2599, Dec. 2014.

[20] X.-Z. Wang *et al.*, "A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1638–1654, Oct. 2015.

[21] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 289–300, Mar. 2002.

[22] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[23] E. Soria-Olivas *et al.*, "BELM: Bayesian extreme learning machine," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 505–509, Mar. 2011.

[24] G.-B. Huang, H. M. Zhou, X. J. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[25] A. A. Mohammed, R. Minhas, Q. M. J. Wu, and M. A. Sid-Ahmed, "Human face recognition based on multidimensional PCA and extreme learning machine," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2588–2597, 2011.

[26] K. A. Toh, "Deterministic neural classification," *Neural Comput.*, vol. 20, no. 6, pp. 1565–1595, 2008.

[27] X. Liu, S. Lin, J. Fang, and Z. Xu, "Is extreme learning machine feasible? A theoretical assessment (part I)," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 7–20, Jan. 2015.

[28] S. Lin, X. Liu, J. Fang, and Z. Xu, "Is extreme learning machine feasible? A theoretical assessment (part II)," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 21–34, Jan. 2015.

[29] J. Cao, K. Zhang, M. Luo, C. Yin, and X. Lai, "Extreme learning machine and adaptive sparse representation for image classification," *Neural Netw.*, vol. 81, no. C, pp. 91–102, Sep. 2016.

[30] J. Cao, J. Hao, X. Lai, C.-M. Vong, and M. Luo, "Ensemble extreme learning machine and sparse representation classification algorithm," *J. Franklin Inst.*, vol. 353, no. 17, pp. 4526–4541, 2016.

[31] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[32] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proc. IEEE Symp. Comput. Intell. Data Min.*, Nashville, TN, USA, 2009, pp. 389–395.

[33] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, nos. 1–3, pp. 359–366, 2008.

[34] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1352–1357, Aug. 2009.

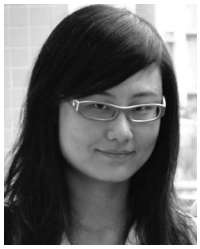
[35] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[36] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Budapest, Hungary, 2004, pp. 985–990.
[37] L. A. Zadeh, "Probability measures of fuzzy events," *J. Math. Anal. Appl.*, vol. 23, no. 2, pp. 421–427, 1968.
[38] A. De Luca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory," *Inf. Control*, vol. 20, no. 4, pp. 301–312, 1972.
[39] G. J. Klir, "Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?" *Fuzzy Set Syst.*, vol. 24, no. 2, pp. 141–160, 1987.
[40] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty and Information*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.
[41] D. Sánchez and E. Trillas, "Measures of fuzziness under different uses of fuzzy sets," in *Advances in Computational Intelligence* (Communications in Computer and Information Science), vol. 298. Heidelberg, Germany: Springer, 2012, pp. 25–34.
[42] F. W. Smith, "Pattern classifier design by linear programming," *IEEE Trans. Comput.*, vol. C-17, no. 4, pp. 367–372, Apr. 1968.
[43] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.
[44] S. Ding, N. Zhang, and J. Zhang, "Unsupervised extreme learning machine with representational features," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 2, pp. 587–595, Apr. 2017s.
[45] S. Balasundaram and D. Gupta, "On optimization based extreme learning machine in primal for regression and classification by functional iterative method," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 5, pp. 707–728, Oct. 2016.
[46] P. Liu, Y. Huang, L. Meng, and S. Gong, "Two-stage extreme learning machine for high-dimensional data," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 5, pp. 765–772, Oct. 2016.
[47] J. Zhang, S. Ding, and N. Zhang, "Incremental extreme learning machine based on deep feature embedded," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 1, pp. 111–120, Feb. 2016.
[48] A. Fu, C. Dong, and L. Wang, "An experimental study on stability and generalization of extreme learning machines," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 1, pp. 129–135, Feb. 2015.

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

From 2001 to 2014, he was a Full Professor and the Dean of the College of Mathematics and Computer Science, Hebei University, Hebei, China. From 1998 to 2001, he was a Research Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. Since 2014, he has been a Full Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include supervised and unsupervised learning, active learning, reinforcement learning, manifold learning, transfer learning, unstructured learning, uncertainty, fuzzy sets and systems, fuzzy measures and integrals, rough set, and learning from big data.

Dr. Wang was a recipient of many awards from the IEEE SMC Society. He is a member of the Board of Governors of the IEEE International Conference on Systems, Man, and Cybernetics (SMC) in 2005, 2007–2009, and 2012–2014, the Chair of the Technical Committee on Computational Intelligence of the IEEE SMC, and a Distinguished Lecturer of the IEEE SMC. He was the Program Co-Chair of the IEEE SMC 2009 and 2010. He is the Editor-in-Chief of the *International Journal of Machine Learning and Cybernetics*. He is also an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, *Information Sciences Journal*, and the *International Journal of Pattern Recognition and Artificial Intelligence*.



Ran Wang (S'09–M'14) received the B.Eng. degree in computer science from the College of Information Science and Technology, Beijing Forestry University, Beijing, China, in 2009, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2014.

From 2014 to 2016, she was a Post-Doctoral Researcher with the Department of Computer Science, City University of Hong Kong. She is currently an Assistant Professor with the College of Mathematics and Statistics, Shenzhen University, Shenzhen, China. Her current research interests include pattern recognition, machine learning, fuzzy sets and fuzzy logic, and their related applications.

Chen Xu received the B.Sc. and M.Sc. degrees from Xidian University, Xi'an, China, in 1986 and 1989, respectively, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, in 1992.

He joined Shenzhen University, Shenzhen, China, in 1992, where he is currently a Professor. From 1999 to 2000, he was a Research Fellow with Kansai University, Suita, Japan, and the University of Hawaii, Honolulu, HI, USA, from 2002 to 2003. His current research interests include image processing, intelligent computing, and wavelet analysis.