PERGAMON

Expert Systems with Applications 23 (2002) 31–37

Expert Systems with Applications

www.elsevier.com/locate/eswa

Short communication

Learning discriminant functions with fuzzy attributes for classification using genetic programming

Been-Chian Chien[a,*], Jung Yi Lin[a], Tzung-Pei Hong[b]

[a]Institute of Information Engineering, I-Shou University, 1, Section 1, Hsueh-Cheng Road, Ta-Hsu Hsiang Kaohsiung County, 84008 Taiwan, ROC
[b]Department of Electrical Engineering, National University of Kaohsiung, 251, Lane 280, Der-Chung Road, Nan-Tzu District, Kaohsiung County, 811 Taiwan, ROC

Abstract

Classification is one of the important tasks in developing expert systems. Most of the previous approaches for classification problem are based on classification rules generated by decision trees. In this paper, we propose a new learning approach based on genetic programming to generate discriminant functions for classifying data. An adaptable incremental learning strategy and a distance-based fitness function are developed to improve the efficiency of genetic programming-based learning process. We first transform attributes of objects into fuzzy attributes and then a set of discriminant functions is generated based on the proposed learning procedure. The set of derived functions with fuzzy attributes gives high accuracy of classification and presents a linear form. Hence, the functions can be transformed into inference rules easily and we can use the rules to provide the building of rule base in an expert system. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Classification; Genetic programming; Knowledge discovery; Fuzzy sets

1. Introduction

Classification is one of the important issues in many applications. A classification problem is that for a given data set, to classify such data set into several predetermined classes by the attributes of data. Owing to the versatility of human activities and unpredictability of data, this mission is difficult and challenged. An accurate classifier can be applied to many applications, such as pattern recognition, disease diagnosis, and business decision-making.

For solving classification problem, many effective and efficient methods have been proposed. Most of them are developed based on mathematical models or theories. For example, the statistical classifiers are built on the Bayesian decision theory (Heckerman & Wellman, 1995). The theory provides a probability model to assign a data into a class that has the highest probability. Although the statistical models are soundness, the main limitation of this approach is that users must have a good knowledge about data properties for performing effective classification. Another well-known method is neural network (Wang, Liu, Hong, & Tseng, 1999). In neural network method, a multi-layered network with m inputs and n outputs is trained with the given training set. We give an input vector to the network, and an n-dimensional output vector is obtained from the outputs of the network. Then the given vector is assigned to the class with the maximum output. The drawbacks of neural network method are that the knowledge representation in the neural network is unclear and the training process is inefficient. The other methods include distance-based classifier and evolutionary approach. Distance-based classifiers, like maximum likelihood classifier (MLC) (Duda & Hart, 1973) and k-nearest neighbor (Duda & Hart, 1973; Han, Karypis, & Kumar, 2001), measure distances among input vectors of data, then classify data into a class with the least distance. Although the main idea of distance-based classifiers is comprehensible, but they are usually time-consuming. The evolutionary approach includes genetic algorithm (GA) (Wang et al., 1999; Wang, Hong, & Tseng, 1998a; Wang, Hong, Tseng, & Liao, 1998b) and genetic programming (GP) (Fretas, 1997; Kishore, Patnaik, Mani, & Agrawal, 2000; Sherrah, Bogner, & Bouzerdoum, 1996). Genetic algorithm encodes classification rules to sequence of bit strings. Bit strings may be replaced by new bit strings after applying the evolutionary operators such as reproduction, crossover, and mutation. After a number of evolving generations, the bit strings with good fitness will be generated. Then, a set of effective classification rules can be obtained from the final set of bit strings satisfying the fitness function. For GP, the classifier can be accomplished in

* Corresponding author.
E-mail addresses: cbc@isu.edu.tw (B.-C. Chien), m893310m@isu.edu.tw (J.Y. Lin), tphong@nuk.edu.tw (T.-P. Hong).

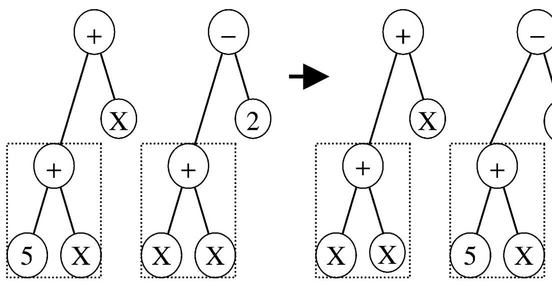

Fig. 1. An example of crossover.

either two ways: classification rules (Fretas, 1997) or classification functions (Kishore et al., 2000). The latter is also called discriminant functions. The main advantage of classifying by discriminant functions instead of classification rules is concise and efficient. However, the discriminant functions are hard to be understood and interpreted since the discriminant functions are usually non-linear. Thus, it is difficult for the user to create the rule base directly in building expert systems.

In this paper, we propose an approach for learning classification functions with fuzzy attributes based on GP. In the proposed approach, we first define a set of fuzzy sets for each numerical attribute to transform the original attributes into fuzzy attributes, then an adaptable incremental learning strategy and a fitness function are proposed to handle effectiveness as well as efficiency in the learning stages of GP. The advantage of transforming attributes by linear fuzzy membership functions is that a set of linear discriminant functions will be learned by the GP learning process and the set of functions can be used to produce the rule base of an expert system easily.

We use the Fisher's Iris data set, a well-known data set in classification problem, to show the feasibility and performance of our proposed method. The experiment also figures out the discriminant functions with original attributes in comparison with the functions with fuzzy attributes to show the contribution of our method.

The remainder of this paper is organized as follows. Section 2 reviews the basic methodology of GP proposed by Koza. In Section 3, we propose a GP-based approach to accomplish the task of effective classification with fuzzy sets. Section 4 shows the creation of classification rules and experimental results. Finally, the conclusions are made in Section 5.

2. Reviews of genetic programming

The technique of GP was proposed by Koza (Koza, 1992; Koza, Golberg, & Fogel, 1996) in 1987. GP has been applied to several applications like symbolic regression, the robot control programs, and classification, etc. GP can discover underlying data relationships and present these relationships by expressions. The expression is constructed

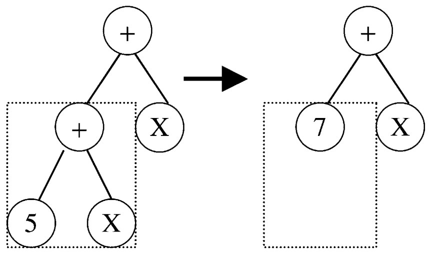

Fig. 2. An example of sub-tree mutation.

by terminals and functions. There are several types of functions can be applied to GP:

1. Arithmetic operations: Addition, subtraction, multiplication and division.
2. Trigonometric functions: Sine and Cosine, etc.
3. Conditional operators and Boolean operators: IF, ELSE and OR, etc.
4. Other add-on operations: Absolution, negative and other user-specific functions.

GP begins with a population that is a set of random created individuals. Each individual represents a potential solution that is represented as a binary tree. Each binary tree is constructed by all possible compositions of the sets of functions and terminals. A fitness value of each tree is calculated by a suitable fitness function. By the fitness value, a set of individuals that with better fitness will be selected. These individuals are used to generate new population in next generation with genetic operators. Genetic operators generally include reproduction, crossover, mutation, and others that are used to evolve functional expressions.

The reproduction operator is the simplest one. After the individuals with better fitness values are selected, this operator reproduces these individuals into the population in next generation. The selected individuals are able to be reserved in offspring by the reproduction operator. The crossover operator needs more actions to generate the new individual. First, select two individuals to be parents. Next, randomly select a sub-tree from each parent and swap the two selected sub-trees. After that, two new individuals are generated. For example, as Fig. 1, there are two individuals $(5 + X) + X$ and $(X + X) - 2$. After the crossover operator is performed, two new individuals $(X + X) + X$ and $(5 + X) - 2$ are created. The mutation operator is usually applied to avoiding local optimum. There are two types of mutation operators: single-node mutation and sub-tree mutation. Single-node mutation is that a terminal or a function in an individual will be replaced by another terminal or function. The other type of mutation, sub-tree mutation, does the same operation with sub-tree instead of single terminal or single function. As the example in Fig. 2, an individual $(5 + X) + X$ becomes $(7 + X)$ by sub-tree mutation operator.

After the evolution of a number of generations, an individual with better fitness value will be obtained. This individual may be the solution that we wanted. However, if the fitness value of such individual still does not satisfy the stop constraint, the process of evolution should be continued until the specified conditions are satisfied.

3. The proposed method for classification

In this section, we describe our proposed method to find discriminant functions for classification using GP. We first give a formal description for classification problem with fuzzy attributes. Then, we describe the training strategy and the fitness function to learn the set of discriminant functions for classification.

3.1. The problem description

At first, the notations of symbols used in this section and a formal description of classification problem with fuzzy attributes are described in the following.

Consider a given data set S, for a data x_j such that $x_j \in S$ having n attributes $A_1, A_2, \ldots, A_n \in \mathbf{R}$. Assume that

$$x_j = (v_{j1}, v_{j2}, \ldots, v_{jn}),$$

where $v_{ji} \in A_i$. Let A_{ik} be one of the fuzzy sets for attribute A_i, and μ_{ik} be the membership function for A_{ik}, $1 \le k \le l_i$, where l_i is the number of fuzzy sets in attribute A_i. The attribute A_i can be transformed to the following fuzzy set:

$$(\mu_{i1}/A_{i1} + \mu_{i2}/A_{i2} + \cdots + \mu_{il_i}/A_{il_i}),$$

where $\mu_{ik} : A_i \to [0,1]$ is the membership function, and $\mu_{ik}(v_{ji})$ is the fuzzy membership value of data x_j for fuzzy set A_{ik}.

Hence, a data x_j is represented to be y_j as the following fuzzy attributes

$$y_j = (\mu_{11}(v_{j1}), \mu_{12}(v_{j1}), \ldots, \mu_{1l1}(v_{j1}),$$

$$\mu_{21}(v_{j2}), \mu_{22}(v_{j2}), \ldots, \mu_{2l2}(v_{j2}), \ldots, \mu_{n1}(v_{jn}), \mu_{n2}(v_{jn}), \ldots,$$

$$\mu_{nln}(v_{jn})).$$

The classification problem with fuzzy attributes is defined as follows:

Assume that there are K predefined classes denoted as $C_1, C_2, \ldots, C_K$. Let $C = \{C_1, C_2, \ldots, C_K\}$, we say that $\langle y_j, c_j \rangle$ transformed from $\langle x_j, c_j \rangle$ is a fuzzy sample of class c_j, $c_j \in C$. We define a fuzzy training set (TS) to be a set of collections of fuzzy samples, as follows:

$$\text{TS} = \{\langle y_j, c_j \rangle | y_j \text{ is fuzzy attributes}, c_j \in C,\ 1 \le j \le m\},$$

where $m = |\text{TS}|$ is the number of fuzzy samples in TS, and m_i is the number of samples belonging to the class C_i, and

$$m = \sum_{i=1}^{K} m_i,\ \ 1 \le i \le K.$$

We define a discriminant function f_i for class C_i,

$$f_i : \mathbf{R}^{n'} \to \mathbf{R},\ \ \text{where } n' = \sum_{i=1}^{n} l_i,$$

such that satisfies the following conditions:

$$\begin{cases} f_i(y_j) \ge a, \text{ if } c_j = C_i \\ f_i(y_j) < a, \text{ if } c_j \ne C_i \end{cases},\ \ \text{where } 1 \le i \le K,\ \ 1 \le j \le m.$$

A set of discriminant functions F for the set of class C is defined by follows,

$$F = \{f_i | f_i : \mathbf{R}^{n'} \to \mathbf{R},\ 1 \le i \le K\}.$$

3.2. The adaptable incremental learning strategy

In the learning procedure, we first prepare the training set TS. The samples in TS usually include positive instances and negative instances. Consider a specified class C_i, a fuzzy sample $\langle y_j, c_j \rangle \in \text{TS}$, $1 \le j \le m$, we say that $\langle y_j, c_j \rangle$ is a positive instance if $c_j = C_i$; otherwise, $\langle y_j, c_j \rangle$ is a negative instance.

After TS is prepared, we use GP to start learning procedure. Conventionally, all of the samples in TS are fed to the learning procedure at a time. However, while the size of TS is getting larger, the training step will relatively spend more time upon learning from samples. In GP, the number of evolving generations will increase rapidly if we want to find an effective solution from a large number of training data set. Thus, for obtaining effective solutions efficiently in GP, we develop an adaptable incremental strategy to proceed to the learning of training set. The main steps are described as follows.

Let g be the specified generations of evolution in each stage of the learning procedure and m' be the number of training samples in each stage. We further define three parameters ρ, α and ω. The ρ is the initial incremental rate of training samples in each stage. The α is the adaptive factor used to adapt incremental rate of samples. The ω is the condition of satisfying the effectiveness of fitness values, $0 \le \rho \le 1$, $0 \le m' \le m$, $\alpha \ge 1$. The main steps are described as follows:

Step 1: Initially, let $\alpha = 1$ and $m' = 0$. We give the condition ω and specify g and ρ.
Step 2: $m' = m' \times \alpha \times \rho + m'$. If $m' \ge m$, then $m' = m$.
Step 3: The population of GP is evolved with the m' training samples for g generations. A function will be obtained after the g generations.
Step 4: If $m' = m$, then stop, else go to next step.
Step 5: If the fitness value satisfies the condition ω, then $\alpha = 2$, else $\alpha = 1$. Go to *Step* 2.

Example. In the Fisher's Iris data set (Fisher, 1936), there are 150 data objects. Assume that the training set TS contains 75 samples in this example. The larger fitness

Table 1
Summary statistic of attributes in Iris set

	Maximum	Mean	Minimum
Sepal length	4.30	5.84	7.90
Sepal width	2.00	3.05	4.40
Petal length	1.00	3.76	6.90
Petal width	0.10	1.20	2.50

value is better and 0 is the largest fitness value of the fitness function.

Stage 1:

Step 1: We initially set $\alpha = 1$, $m' = 0$, and $m = 75$. We also specify $g = 1000$ and $\rho = 0.2$. The condition of ω is 'the fitness value equals to 0'.
Step 2: $m' = m \times \alpha \times \rho = 75 \times 1 \times 0.2 = 15$.
Step 3: The population of GP is evolved with $m' = 15$ training samples for $g = 1000$ generations. After that, a function is obtained.
Step 4: Since $m' = 15 \neq m = 75$, go to next step.
Step 5: If the fitness value is 0 at this time, then $\alpha = 2$. The procedure goes to *Step* 2 and begins the next stage.

Stage 2:

Step 2: $m' = m \times \alpha \times \rho + m' = 75 \times 2 \times 0.2 + 15 = 45$.
Step 3: The population of GP is evolved with $m' = 45$ training samples for $g = 1000$ generations. We get a new function.
Step 4: Since $m' = 45 \neq m = 75$, go to next step.
Step 5: If the fitness value is not 0 at this time, the procedure continues the next stage by $\alpha = 1$.

Stage 3:

Step 2: $m' = m \times \alpha \times \rho + m' = 75 \times 1 \times 0.2 + 45 = 60$.
Step 3: The population of GP is evolved with $m' = 60$ training samples for $g = 1000$ generations, then a new function is obtained.
Step 4: Since $m' = 45 \neq m = 75$, go to next step.
Step 5: If the fitness value is not 0 at this time, the procedure continues the next stage by $\alpha = 1$.

Stage 4:

Step 2: $m' = m \times \alpha \times \rho + m' = 75 \times 1 \times 0.2 + 60 = 75$.
Step 3: The population of GP is evolved with $m' = 75$ training samples for $g = 1000$ generations.
Step 4: Since $m' = 75 \geq m = 75$, the learning procedure is stopped and the obtained function is the discriminant function.

3.3. The fitness function

The fitness value is derived from a predefined fitness function, which is used to evaluate the effectiveness of an individual. The fitness function of our classification method is defined as follows.

We consider a discriminant function f_i of a class C_i and a specified constant a. For the positive instance, we urge $f_i(y_j) \geq a$, and for negative instance, $f_i(y_j) < a$. To achieve the objectivity of f_i, two parameters p and q are defined and let $p > a$, $q < a$, and $p + q = 2 \times a$. We then define two measures of error degrees for positive instances and negative instances, respectively.

The error degree for a positive instance is defined as

$$D_\mathrm{p} = \begin{cases} 0, & \text{if } c_j = C_i \text{ and } f_i(y_j) \geq a, \\ [p - f_i(y_j)]^2, & \text{if } c_j = C_i \text{ and } f_i(y_j) < a, \end{cases}$$

and the error degree for a negative instance is defined as

$$D_\mathrm{n} = \begin{cases} 0, & \text{if } c_j \neq C_i \text{ and } f_i(y_j) < a, \\ [f_i(y_j) - q]^2, & \text{if } c_j \neq C_i \text{ and } f_i(y_j) \geq a, \end{cases}$$

Thus, the fitness function is defined as

$$-\sum_{j=1}^{m'} (D_\mathrm{p} + D_\mathrm{n}),$$

where m' is the number of training samples in the current learning stage, $\langle y_j, c_j \rangle \in \mathrm{TS}$, $1 \leq j \leq m'$. Since the fitness value of an individual is represented by the degree of errors after evaluating all training samples, we urge that the fitness value approaches to zero for producing an effective function for classification.

4. Deriving classification rules and experiments

We take the Fisher's Iris data set (Fisher, 1936), a well-known benchmark for classification problem, as an example to show the discovery of classification rules from classification functions. At first, we describe the transforming method of fuzzy attributes. Then, we generate the classification rules of Iris data. At last, we define the measures of accuracy and show the performance of the proposed approach.

4.1. Fisher's iris data set

We first define the proper fuzzy membership functions for Iris data set. In Fisher's Iris data set (Fisher, 1936), there are four numerical attributes including sepal length, sepal width, petal length and petal width, which are denoted as *sl*, *sw*, *pl* and *pw*, respectively. We give three linguistic terms, 'short', 'medium' and 'long', as the fuzzy sets for each attribute. The fuzzy sets are *sl_L*, *sl_M*, *sl_S*, *sw_L*, *sw_M*, *sw_S*, *pl_L*, *pl_M*, *pl_S*, *pw_L*, *pw_M*, and *pw_S*. In order to construct the fuzzy membership functions, we calculate the statistical information about each attribute as shown in Table 1. In our experiment, the maximum, minimum, and mean are directly used to establish the triangular

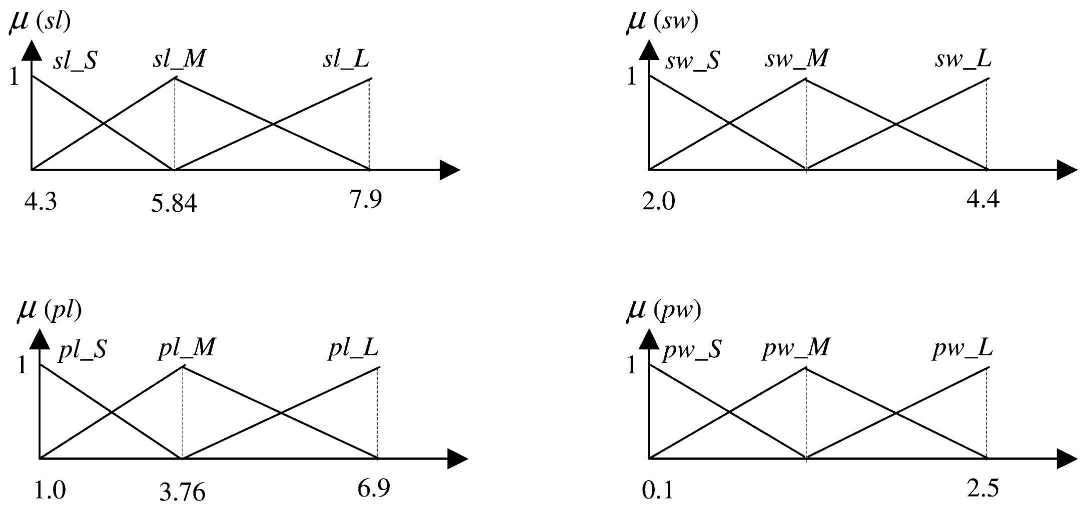

Fig. 3. The membership functions for Iris dataset.

membership functions for each fuzzy set. The membership functions for the four attributes of Iris data are shown in Fig. 3.

4.2. Classification rules

In Fisher's Iris (Fisher, 1936) data set, there are 150 samples separated into three classes: 'setosa', 'versicolor' and 'virginica'. Each class contains 50 samples. We randomly select 25 data from each class to construct the training set TS. After fuzzy membership functions in Section 4.1 are applied, the data is represented by twelve fuzzy attributes: *sl_L*, *sl_M*, *sl_S*, *sw_L*, *sw_M*, *sw_S*, *pl_L*, *pl_M*, *pl_S*, *pw_L*, *pw_M*, and *pw_S*.

The parameters used in the adaptable incremental learning strategy are $g = 1000$, $\rho = 0.2$, $\alpha = 2$ and the condition of fitness value is set to be zero. The parameters used in fitness function are $p = 100$, $q = -100$ and $a = 0$. The parameters used in GP program (Singleton, 1994) are summarized in Table 2.

Table 2
Parameters used in the experiment of Iris

Parameters	Values
Node, constant mutate weight	0.435
Mutation weight annealing	0.40
Incremental rate ρ	0.2
Population size	1000
Crossover weight	0.28
Crossover weight annealing	0.20
Adaptive factor α	2
Condition of fitness value	0
Generations per stage g	1000
Function set	$+, -, \times, \div$
Selection method	Tournament
Tournament size	7
Terminal set	*sl_L, sl_M, sl_S, sw_L, sw_M, sw_S, pl_L, pl_M, pl_S, pw_L, pw_M, pw_S*

After the GP learning procedure, three linear discriminant functions are obtained as follows:

$$f_{\text{setosa}} = (sl\_S - pl\_M + sw\_L - pl\_L)$$

$$f_{\text{versicolor}} = (29 \times pw\_M - sl\_M + 71 \times pl\_M - 67)$$

$$f_{\text{virginica}} = (2 \times pw\_L + pl\_L - sw\_L - sl\_L - pl\_M).$$

The functions are linear and it is easy for us to transform them into to a set of rules. Since $a = 0$, we get the following three rules:

IF$(sl\_S + sw\_L \geq pl\_M + pl\_L)$
THEN setosa.
IF$(29 \times pw\_M + 71 \times pl\_M \geq 67 + sl\_M)$
THEN versicolor.
IF$(2 \times pw\_L + pl\_L \geq sw\_L + sl\_L + pl\_M)$
THEN virginica.

We compare the above discriminant functions with the functions obtained by original attributes in the Fisher's Iris data set. The three discriminant functions obtained by original attributes are:

$$f'_{\text{setosa}} = sw - pl$$

$$f'_{\text{versicolor}} = \frac{pl \times sl - \dfrac{pl}{-119} - \dfrac{-4 \times (pw - sl)}{pl}}{\dfrac{pw \times sl}{pl - sl} + 5 + sw + \dfrac{-16}{-104}}$$

$$f'_{\text{virginica}} = \frac{115}{29 \times pw + sl - 53}.$$

Obviously, the discriminant functions obtained by *sl*, *sw*, *pl* and *pw* are non-linear and it is difficult for us to understand.

Table 3
Experimental results with fuzzy attributes

	Fuzzy attributes		
	f_{setosa}	$f_{\text{versicolor}}$	$f_{\text{virginica}}$
Setosa	25	0	0
Versicolor	0	23	2
Virginica	0	0	25
Precision (%)	100	100	92.6
Recall (%)	100	92.0	100
Average accuracy (%)	97.3		
Overall accuracy (%)	97.3		

4.3. Evaluations of performance

The traditional evaluation on classification usually consists of average accuracy and overall accuracy (Wang et al., 1999). However, since the task of our proposed approach is accomplished by different discriminant function f_i for each class C_i, we use two additional measures, precision and recall, to evaluate the accuracy of each discriminant function f_i as well as the two traditional measures.

We define the precision and recall in the following. Let N_i be the number of objects in class C_i, N_{f_i} be the number of objects recognized by function f_i and $N_{f_i}^{\ i}$ be the number of objects belonging to class C_i and recognized by function f_i. For a specific class C_i, the precision and recall for the corresponding discriminant function f_i are defined as Eqs. (1) and (2), respectively.

$$\text{Precision} = \frac{N_{f_i}^i}{N_{f_i}} \tag{1}$$

$$\text{Recall} = \frac{N_{f_i}^i}{N_i}. \tag{2}$$

Assume that $|S|$ is the number of objects included in the set of test data S. K is the number of classes. The definitions of average accuracy and overall accuracy (Wang et al., 1999) are given in Eqs. (3) and (4).

$$\text{Average accuracy} = \frac{1}{K}\sum_{i=1}^{K} \text{the recall of } f_i \tag{3}$$

Table 4
Experiment results with original attributes

	Original attributes		
	f'_{setosa}	$f'_{\text{versicolor}}$	$f'_{\text{virginica}}$
Setosa	25	0	0
Versicolor	0	23	1
Virginica	0	0	23
Precision (%)	100	100	95.8
Recall (%)	100	92.0	92.0
Average accuracy (%)	94.7		
Overall accuracy	94.7		

Table 5
Accuracy comparison

Models or methods	Testing recognition rate (%)
GVS (Hong & Tseng, 1997)	96.0
FEBFC with 4 features (Lee et al., 2001)	96.7
FRG with GA (Lin & Chen, 2001)	96.87
FEBFC with 2 selected features (Lee et al., 2001)	97.1
FIL (Wang et al., 1999)	97.3
Our method	97.3

$$\text{Overall accuracy} = \frac{\sum_{i=1}^{K} N_{f_i}^i}{|S|}. \tag{4}$$

The four different measures stand for distinct meanings of accuracy. A discriminant function f_i with higher precision has lower misclassification rate for class C_i. A discriminant function f_i with higher recall means that the f_i behaves higher recognition rate for class C_i. Average accuracy stands for the average recognition rate of all discriminant functions. Overall accuracy presents the performance of recognition for a classifier.

The performance of f_{setosa}, $f_{\text{versicolor}}$, and $f_{\text{virginica}}$ are shown in Table 3. The results are done by 75 test data exclude the 75 data in training set. We found that the discriminant functions obtain high accuracy either in precision or in recall. The high accuracy shows that the proposed classification method is accurate and effective. Comparing with the results of f'_{setosa}, $f'_{\text{versicolor}}$, and $f'_{\text{virginica}}$ in Table 4, the results of f_{setosa}, $f_{\text{versicolor}}$, and $f_{\text{virginica}}$ are more accurate than f'_{setosa}, $f'_{\text{versicolor}}$, and $f'_{\text{virginica}}$. Although the precision in $f'_{\text{virginica}}$ is higher than $f_{\text{virginica}}$, two objects are not recognized and lost. $f_{\text{virginica}}$ can recognize all objects belonging to virginica, hence, we get better average accuracy and overall accuracy. We compare the results with previous works in (Hong & Tseng, 1997; Lee, Chen, Chen, & Jou, 2001; Lin & Chen, 2001; Wang et al., 1999) in Table 5, the accuracy of proposed method is better than the other methods.

Furthermore, we found that the learning time of the experiment is fast. For each class of Iris dataset, the adaptable incremental learning strategy never spent more than 1 min. It could be used to show that the proposed learning strategy is effective .

5. Conclusions

This paper presents an efficient algorithm to generate the discriminant functions for fuzzy data classification based on GP. The main method includes the adaptive incremental learning strategy and a fitness function. The experiments show that the proposed method has higher accuracy in the Fisher's Iris data set. We compared the experimental results with original Fisher's Iris data set and found that the obtained discriminant functions with fuzzy attributes are more concise and easier than the original attributes to transform into classification rules. Thus, our method can present more knowledge than the approach of neural network. The research of classification based on GP is a new direction. There are relatively few works in the issue of classification using GP. It is worth to investigate the related problems. In the future, we will focus on the classification problem with miss values in the data set.

References

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*, New York: Wiley.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, Part II*, *7*, 179–188.

Fretas, A. A. (1997). A genetic programming framework for two data mining tasks: classification and generalized rule induction. *Proceedings on Second Annual Conference* Morgan Kaufmann (pp. 96–101).

Han, E. H., Karypis, G., & Kumar, V. (2001). *Text categorization using weight adjusted k-nearest neighbor classification*. The Fifth Pacific Asia Conference on Knowledge Discovery and Data Mining (pp. 53–65).

Heckerman, D., & Wellman, M. P. (1995). Bayesian networks. *Communications of the ACM*, *38* (3), 27–30.

Hong, T. P., & Tseng, S. S. (1997). A generalized version space learning algorithm for noisy and uncertain data. *IEEE Transactions on Knowledge Data Engineering*, *9*, 336–340.

Kishore, J. K., Patnaik, L. M., Mani, V., & Agrawal, V. K. (2000). Application of genetic programming for multicategory pattern classification. *IEEE Transactions on Evolutionary Computation*, *4* (3), 242–258.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*, Cambridge: MIT Press.

Koza, J. R., Goldberg, D. E., & Fogel, D. B. (Eds) (1996). *Genetic Programming 1996*. Cambridge: MITPress.

Lee, H. M., Chen, C. M., Chen, J. M., & Jou, Y. L. (2001). An efficient fuzzy classifier with feature selection based on fuzzy entropy. *IEEE Transactions on Systems, Man, and Cybernetics—Part b: Cybernetics*, *31* (3), 426–432.

Lin, H. L., & Chen, S. M. (2001). A new method for generating weighted fuzzy rules from training instances using genetic algorithms. *Proceedings of sixth conference on artificial intelligence and applications* (pp. 628–633).

Sherrah, J., Bogner, R. E., & Bouzerdoum, A. (1996). Automatic selection of features for classification using genetic programming. *Proceedings on Australian New Zealand Conference on Intelligent Information Systems* (pp. 284–287). New Zealand.

Singleton, A. (1994). Genetic programming with C++ (pp. 171–176). Byte, February.

Wang, C. H., Hong, T. P., & Tseng, S. S. (1998a). Integrating fuzzy knowledge by genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *2* (4), 138–149.

Wang, C. H., Hong, T. P., Tseng, S. S., & Liao, C. M. (1998b). Automatically integrating multiple rule sets in a distributed-knowledge environment. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, *28* (3), 471–476.

Wang, C. H., Liu, J. F., Hong, T. P., & Tseng, S. S. (1999). A fuzzy inductive learning strategy for modular rules. *Fuzzy Set and Systems*, *103*, 91–105.