ABSTRACT

Knowledge discovery and data mining have become a hot research topic of late years. Classification is one of the most important problems in knowledge discovery. So many different classification algorithms have been developed for classifying data. In this paper, we present an effective scheme for classifying data with multi-category based on the technique of genetic programming. The proposed method presents a training strategy called adaptable incremental learning strategy and a well-defined fitness function to learn the set of discriminating functions from the given examples using genetic programming. For a $k$-class problem, after the $k$ discriminating functions are generated, a Z-value measure is developed and used to resolve the problem of conflict. The experiments show that the proposed GP-based classification algorithm is efficient and has high accuracy.

(Keywords: Knowledge discovery, Data mining, Genetic programming, Classification)

1. INTRODUCTION

Developing an effective and effective knowledge discovery system is a challenge for the research area of machine learning and data mining. The purpose of machine learning is to try to find human knowledge from a given data set, and the goal of data mining is to make chaotic data turn into valuable information. Since the versatility of human activities and the unpredictability of data, this challenge has motivated so many research issues. Classification is one of the major issues. The task of classification is that a new object needs to be assigned to a class from a predefined set of classes based on the observed attributes of the objects. Many applications can be treated as extensions of classification problem. For example, pattern recognition, disease diagnosis, and business decision-making.

Many different methods have been proposed for classification problem in the past decades. Most of the previous methods are based on some mathematical models or theories. For example, the statistical classifiers are built on the Bayesian decision theory [20]. The theory provides a probability model for classification by minimizing the probability of total misclassification rate. 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 [20]. In neural network method, a multi-layered network with $m$ inputs and $n$ outputs is trained with a given set of training data. 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 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 classifiers and evolutionary approaches. Distance-based classifiers, like maximum likelihood classifier (MLC) [2] and $k$-nearest neighbor [2][5], evaluate distances among input vectors of objects, then classify objects into the individual class with the smallest distance. There are several disadvantages for distance-based classifiers. For example, the distance-based classifiers can only work well under the data set with normal distribution and they are usually time-consuming. The evolutionary approaches, generally, include genetic algorithm (GA) [7][8][9] and genetic programming (GP) [4][10][16]. Genetic algorithm encodes classification rules to be sequence of bit strings called genes. The evolution operations such as reproduction, crossover and mutation then are used to generate the next generation of classification rules with better fitness. After the specified number of generations is computed or the conditions of fitness functions are satisfied, a set of effective classification rules can be obtained usually. For genetic programming, the classification is generally accomplished in another way: the discriminating functions[10]. The main advantage of classifying by discriminating functions instead of classification rules is concise and efficient. However, the situation of conflicts may be happened while classifying objects by the discriminating functions. A conflict occurs when an object is assigned to more than two different classes by distinct discriminating functions at the same time. To resolve the problem of conflict, Kishore proposed a method called the strength of association measure (SA measure) [10]. The SA measure calculates the ratio of correct classified examples in the set of training data to be a SA value for each discriminating function. While a conflict occurs in classifying, the conflicting object is assigned to the class with higher SA value. Unfortunately, the discriminating function with a higher SA may swamps other functions with a lower SA, misclassification will occur in such situation [10]. Since the SA measure is not accurate enough, an effective adjustment method is needed.

In this paper, we propose a new approach based on genetic programming for multi-category classification. In the proposed approach, we give an adaptable incremental
learning strategy to handle effectiveness as well as efficiency in learning stage of genetic programming. We also design a new fitness function for dealing with the examples in order to obtain the corresponding discriminating function for each class. For resolving the problem of conflicting, the Z-value measure method is proposed. Two well-known data sets are selected to show the performance of the proposed method in our experiments: the Wisconsin diagnostic breast cancer (WDBC) and the Fisher’s Iris data sets. We also give a comparison of our method with the other methods. The results turn out that the discriminating functions obtained by the proposed method for classifying are efficient and effective. Furthermore, our GP-based learning algorithm is more efficient than neural network algorithm and more stable than the previous GP-based methods.

The remainder of this paper is organized as follows. Section 2 reviews the basic methodology of genetic programming proposed by Koza. In Section 3, we propose a GP-based approach and a conflict resolution method to accomplish the task of effective classification. Section 4 describes the experimental results and gives a comparison with previous method. Finally, conclusions are made in Section 5.

2. GENETIC PROGRAMMING

The technique of genetic programming (GP) was proposed by Koza [12][13] about in 1990’s. Genetic programming has been applied to a wide range of areas, such as symbolic regression, the evolution of robot control programs and the evolution of classification, etc. Genetic programming can discover underlying data relationships and presents these relationships by expressions. The expression is constructed by terminals and functions. There are several types of functions can be applied for genetic programming, for example, Arithmetic operations, trigonometric functions, Conditional operators and Boolean operators, etc.

Genetic programming begins with a set of randomly created individuals called population. 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 suitable fitness function should be given for evaluating the fitness of each binary tree. Then, a set of individuals with better fitness will be selected and used to generate the population of 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. This operator copies the individuals with better fitness values to the population of next generation directly. The individuals with better fitness values will be able to be kept continuously in offspring by the reproduction operator.

The crossover operator needs more actions to generate the new individual. First, two individuals are picked out as parents. Then two sub-trees are randomly selected, respectively, from the parents and swapped each other. After that, two new individuals are generated.

The mutation operator is usually used for avoiding local optimum. There are two types of mutation operator: Single-node mutation and sub-tree mutation. Single-node mutation is that a terminal or a function in an individual can be replaced by another terminal or function. The other type of mutation, sub-tree mutation, does the same operation with sub-tree instead of a terminal or a function. After the evolution of a number of generations, an individual with fine fitness value usually will be contained in the population. We can take the one with the best fitness value as our result at that time. However, if the fitness values are still not satisfied, the process of evolution could be continued until the specified conditions are reached.

3. THE PROPOSED METHOD

In this section, we describe our proposed method to find discriminating functions for classification using genetic programming. We first give a formal description for classification problem. Then, we describe an adaptable incremental learning strategy for training the given samples effectively and efficiently. Furthermore, we give a new fitness function and provide the algorithm to find the appropriate discriminating functions for classification. Finally, a conflicting resolution, Z-value measure, is contributed to classify ambiguous objects.

3.1 Formal Description of Classification Problem

The notations of used symbols and a formal description for classification problem are described in the following. Consider a given data set \( S \) for each data \( x_i \in S \) having \( n \) attributes:

\[
x_i = (v_{i1}, v_{i2}, \ldots, v_{in}), \quad 1 \leq i \leq n,
\]

where \( v_{ij} \in \mathbb{R} \) stands for the \( j \)-th attribute of \( x_i \). Assume that there is \( K \) predefined classes, \( C_1, C_2, \ldots, C_K \). Let \( C = \{C_i, C_2, \ldots, C_K\} \), we say that \( <x_i, c_j> \) is a sample if the data \( x_i \) has been assigned to a specific class \( c_j \). We define a training set (TS) to be a set of samples, as follows:

\[
TS = \{<x_i, c_j> | x_i = (v_{i1}, v_{i2}, \ldots, v_{in}), c_j \in C, 1 \leq j \leq m\}
\]

where \( m \) is the number of samples in TS, denoted as \(|TS| = m\). Let \( m = (m_1 + m_2 + \ldots + m_1 + \ldots + m_K) \), where \( m_i \) is the number of samples belonging to the class \( C_i \) in TS, \( 1 \leq i \leq K \). A discriminating function \( f_i \) is a function mapping from \( \mathbb{R}^n \) to \( \mathbb{R} \). For a sample \( <x_i, c_j> \) should satisfy the following conditions,

\[
\begin{align*}
&f(x_i) \geq a, \quad \text{if } c_j = C_i \\
&f(x_i) < a, \quad \text{if } c_j \neq C_i,
\end{align*}
\]

where \( 1 \leq i \leq K, 1 \leq j \leq m \).

A set of discriminating functions \( F \) is defined by follows,

\[
F = \{ f_i \} | f_i : \mathbb{R}^n \rightarrow \mathbb{R}, 1 \leq i \leq K \}.
\]

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 sample \( <x_i, c_j> \in TS, 1 \leq j \leq m \), we say that \( <x_i, c_j> \) is a positive instance if \( c_j = C_i \) otherwise, \( <x_i, c_j> \) is a negative instance.
After TS is prepared, we start the learning procedure using genetic programming. 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 genetic programming, the number of evolving generations will increase rapidly if we want to find an effective solution in a large training set. Thus, for obtaining effective solutions efficiently in genetic programming, we proceed to learn from the training set using the adaptable incremental strategy. Assume that \( g \) is the specified generation of evolution in each stage of the learning procedure, and \( m' \) be the number of training samples in each stage, \( m' \leq m \). We further define three parameters \( \rho \), \( \alpha \) and \( \omega \). The \( \rho \) is the basic incremental rate of training samples in each stage. The \( \alpha \) is the adaptive factor used to adapt the rate of incremental samples. The \( \omega \) is the specified criterion of fitness value used to determine if the incremental rate of training samples should be double, where \( 0 \leq \rho \leq 1 \), \( 0 \leq m' \leq m \), \( \alpha \geq 1 \). While the fitness value satisfies the criterion of \( \omega \), the training samples will be incremented doubly in the next stage. The main training steps are described in the following.

Step 1: Initially, we specified \( g \), \( \rho \), \( \alpha \), \( \omega \) = 1 and \( m' = 0 \).

Step 2: \( m' = m \times \alpha \times \rho + m' \). If \( m' \geq m \), then \( m' = m \).

Step 3: Generating the population and evolving the \( m' \) training samples for \( g \) generations with genetic operations.

Step 4: If \( m' = m \), then stop, otherwise go to Step 5.

Step 5: Evaluating a target function with the best fitness value from the population.

Step 6: If the best fitness value satisfies \( \omega \), then \( \alpha = 2 \), otherwise \( \alpha = 1 \). Go to Step 2.

### 3.3 The Fitness Function

The fitness value is obtained by a predefined fitness function therefore is important to evaluate the valuable of an individual. The fitness function of our classification method is defined as follows.

We consider a discriminating function \( f_i \) of a class \( C_i \) and a specified constant \( a \). We urge that \( f_i(x) > a \) for the positive instance and \( f_i(x) \leq a \) for negative instance. To achieve the objectivity of \( f_i \), we define two parameters \( p \) and \( q \), let \( p > a \), \( q < a \) and \( p + q = 2 \times a \). We define the error measure of a positive instance to be

\[
D_p = \begin{cases} 
0 & \text{if } c_j = C_i \text{ and } f_i(x) \geq a \\
(p - f_i(x))^2 & \text{if } c_j = C_i \text{ and } f_i(x) < a 
\end{cases}
\]

and the error measure of a negative instance to be

\[
D_n = \begin{cases} 
0 & \text{if } c_j \neq C_i \text{ and } f_i(x) < a \\
(f_i(x) - q)^2 & \text{if } c_j \neq C_i \text{ and } f_i(x) \geq a 
\end{cases}
\]

Defined the fitness value of an individual by

\[
\sum_{j=1}^{m'} (D_p + D_n)
\]

where \( m' \) is the number of training samples in the current learning stage, \( < c_j, x_j > \in TS \), \( 1 \leq j \leq m' \). Since we use the negative of errors to be the fitness of an individual, the best fitness value is zero. While the fitness value of a function \( f_i \) is zero, it means that the function \( f_i \) can discriminate the samples of class \( C_i \) from the other classes in TS. Hence, the function \( f_i \) can be used to be the discriminating function of class \( C_i \).

### 3.4 Z-value Measure

In general, classifiers cannot recognize all objects completely correct in real applications. Except the case of misclassification, two conflicting situations may occur.

1. A recognized object is classified into two or more classes at the same time.
2. An object does not belong to any class after classified by a classifier.

The above two situations are called conflicts. A complete classifier usually includes a refined method used to resolve conflicts. In our approach, we provide a resolution, named Z-value measure, described as follows.

For a discriminating function \( f_i \in F \) and samples \( <c_j, x_j>\in TS \) with \( c_j = C_i \), let \( \mu_i \) be the mean of values of \( f_i(x_j) \), \( 1 \leq j \leq m_i \). That is,

\[
\mu_i = \frac{\sum_{<c_j, x_j> \in TS, c_j = C_i} f_i(x_j)}{m_i}, 1 \leq j \leq m_i, 1 \leq i \leq K.
\]

For each \( \mu_i \), the standard deviation of values of \( f_i(x_j) \), \( 1 \leq j \leq m_i \), is defined as

\[
\sigma_i = \sqrt{\frac{\sum_{<c_j, x_j> \in TS, c_j = C_i} (f_i(x_j) - \mu_i)^2}{m_i}}, 1 \leq j \leq m_i, 1 \leq i \leq K.
\]

For a given data \( x_j \in S \) and a discriminating function \( f_i \), where \( S \) is the given data set, the Z-value of data \( x_j \) for \( f_i \) is defined as

\[
Z(x_j) = \frac{|f_i(x_j) - \mu_i|}{\sigma_i},
\]

where, \( 1 \leq j \leq |S|, 1 \leq i \leq K \). If the discriminating functions in \( F \) can determine the class of an object \( x_j \) uniquely, we complete the classifying task. However, once the object is not recognized by any discriminating function or the object is recognized by more than two discriminating functions in \( F \). The Z-value measure will be applied to determine to which class the object should be assigned. The procedure of conflict resolution using Z-value measure is shown in the following.

Step 1: Initially, \( i = 1 \) and there exists a set \( Z \) such that \( Z = \varnothing \).

Step 2: If \( f_i(x_j) \geq 0 \), that is, data \( x_j \) is recognized by \( f_i \), then \( Z = \{ f_i \} \cup Z \).

Step 3: If \( i < K \), then \( i = i + 1 \), go to Step 2. Otherwise, go to Step 4.

Step 4: Let \( |Z| \) be the number of functions in \( Z \). If \( |Z| = 1 \), the unique class \( C_i \) corresponding to the function \( f_i \) in \( Z \) will be returned and stop; otherwise, go to Step 5.

Step 5: If \( |Z| = 0 \), then \( Z = F \).

Step 6: Compute \( Z(x_j) \), where \( f_i \in Z \).

Step 7: Find \( k = \arg \min_{f_i \in Z} |Z(x_j)| \), the data \( x_j \) will be assigned to the class \( C_k \).
4. EXPERIMENTAL RESULTS

The experiments are done by using a PC with 866MHz CPU and 128MB RAM. We modify the GP Quick 2.1[18] to fit the requirements of our proposed method and perform the experiments. The test data sets include the Wisconsin Breast Cancer data set [17] and Iris data set [3]. The two data sets are well-known benchmark for evaluating the performance of classifiers.

Since each discriminating function fi is used to recognize its corresponding class Ci, for demonstrating the effectiveness of each discriminating function, the performance is evaluated by two additional measures: precision and recall, except the traditional measures of average accuracy and overall accuracy [19]. We give the definitions of measures as follows: For a specific class Ci, the precision and recall of its corresponding discriminating function fi are defined as following. Let Ni be the number of objects belonging to class Ci, Nfi be the number of objects recognized by function fi and N'i be the number of objects belonging to class Ci and recognized by function fi. For a specific class Ci, the precision and recall of its corresponding discriminating function fi are defined as equations (1) and (2), respectively.

\[
\text{Precision} = \frac{N'_i}{N_fi} \quad (1)
\]

\[
\text{Recall} = \frac{N'_i}{N_i} \quad (2)
\]

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 [19] are given in equations (3) and (4).

\[
\text{Average accuracy} = \frac{1}{K} \sum_{i=1}^{K} \text{the recall of } f_i \quad (3)
\]

\[
\text{Overall accuracy} = \frac{\sum_{i=1}^{K} N'_i}{|S|} \quad (4)
\]

A discriminating function fi with higher precision has lower misclassification rate for class Ci. A discriminating function fi with higher recall means that the fi behaves higher recognition rate for class Ci. The average accuracy stands for the average recognition rate of discriminating functions. The overall accuracy presents the performance of recognition for a classifier.

I. Wisconsin Diagnostic Breast Cancer data set

The first experiment uses Wisconsin diagnostic breast cancer (WDBC) data set [17]. There are 569 data in WDBC separated into two classes: “malignant” and “benign”. There are 212 data in the class of “malignant” and 357 data in the class of “benign”. Each data of WDBC data set contains 30 numerical attributes. We randomly select 106 data from “malignant” and 177 data from “benign” to be the training set. Therefore, the total number of training set 7S is 283 samples. The remainders are used to be test set.

The parameters used in adaptable incremental training strategy are \(g=1000\), \(\rho=0.1\), \(\sigma=0\) and \(\alpha=2\). The parameters of fitness function are set to be \(p=100\), \(q=-100\) and \(a=0\). The 30 attributes of a sample are denoted as \(F1, F2, \ldots, F30\), which are the terminals in GP.

The experimental results for WDBC data are shown in Table 1 and Table 2. Table 1 is the results without Z-value measure and Table 2 is the results with Z-value measure. Furthermore, since a two-class classification problem can be finished by a single function like \(f_{malignant}\) or \(f_{benign}\), we also show the results classified by a single discriminating function. In Table 3, assume that we use \(f_{malignant}\) to classify objects, and if an object were not assigned to the class “Malignant” by \(f_{malignant}\) it would be assigned to the “benign” class directly (denoted as \(-f_{malignant}\)). The similar evaluation is done for \(f_{benign}\) in the right column of Table 4. We found that the results in either the average accuracy or the overall accuracy are improved by Z-value measure. Furthermore, the results of using Z-value measure are better than the results using single discriminating function.

II. Fisher’s Iris

The second experiment uses Fisher’s Iris data set [3]. There are 150 data separated into three classes: “setosa”, “versicolor” and “virginica”. Each class contains 50 data. There are four numerical attributes namely sepal length, sepal width, petal length and petal width. We denote the four attributes as \(SL, SW, PL, PW\), respectively.

We randomly select 25 data from each class to construct the training set. The parameters used in the adaptable incremental learning strategy are \(g=1000\), \(\rho=0.2\), \(\sigma=0\) and \(\alpha=2\). The parameters of fitness function are \(p=100\), \(q=-100\) and \(a=0\).

After the learning procedure, three discriminating functions are obtained:

\[
f_{setosa} = SW-PL; \quad f_{versicolor} = PL \times SL - \frac{PL \times SL \times (PW-SL)}{-119} \quad f_{virginica} = \frac{PW \times SL}{PL-SL} + 5 + SW + \frac{16}{-104}
\]

The performance of \(f_{setosa}, f_{versicolor}\) and \(f_{virginica}\) are shown in Table 4, which is the result without Z-value measure, and Table 5, which is the result with Z-value measure. We found that the method of Z-value measure also improves the accuracy either in precision or in recall. Finally, we compare with the results of Kishore [10] in Table 6, where \(f_{setosa}, f_{versicolor}\) and \(f_{virginica}\) are the discriminating functions of them. The proposed method clearly behaves better accuracy than Kishore [10].

<table>
<thead>
<tr>
<th>Table 1: WDBC, without Z-value measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test set only</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Malignant</td>
</tr>
<tr>
<td>f_{malignant}</td>
</tr>
<tr>
<td>99</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>98.0%</td>
</tr>
<tr>
<td>93.4%</td>
</tr>
</tbody>
</table>
Table 2: The WDBC results with Z-value measure

<table>
<thead>
<tr>
<th></th>
<th>Test set only</th>
<th>All WDBC data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>f_malignant</td>
<td>f_benign</td>
</tr>
<tr>
<td>Malignant</td>
<td>103</td>
<td>3</td>
</tr>
<tr>
<td>Benign</td>
<td>6</td>
<td>174</td>
</tr>
<tr>
<td>Precision</td>
<td>94.5%</td>
<td>98.3%</td>
</tr>
<tr>
<td>Recall</td>
<td>97.2%</td>
<td>96.7%</td>
</tr>
<tr>
<td>Average</td>
<td>96.9%</td>
<td>97.2%</td>
</tr>
<tr>
<td>Overall</td>
<td>96.9%</td>
<td>97.4%</td>
</tr>
</tbody>
</table>

Table 3: WDBC, single discriminating function used

<table>
<thead>
<tr>
<th></th>
<th>Test set only</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>f_malignant</td>
</tr>
<tr>
<td>Malignant</td>
<td>99</td>
</tr>
<tr>
<td>Benign</td>
<td>2</td>
</tr>
<tr>
<td>Precision</td>
<td>98.0%</td>
</tr>
<tr>
<td>Recall</td>
<td>93.4%</td>
</tr>
<tr>
<td>Average</td>
<td>96.1%</td>
</tr>
<tr>
<td>Overall</td>
<td>96.9%</td>
</tr>
</tbody>
</table>

Table 4: The Iris results without Z-value measure

<table>
<thead>
<tr>
<th></th>
<th>f_setosa</th>
<th>f_versicolor</th>
<th>f_virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Versicolor</td>
<td>0</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>Virginica</td>
<td>0</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>Precision</td>
<td>100%</td>
<td>100%</td>
<td>95.8%</td>
</tr>
<tr>
<td>Recall</td>
<td>100%</td>
<td>92.0%</td>
<td>92.0%</td>
</tr>
</tbody>
</table>

Table 5: The Iris results with Z-value measure

<table>
<thead>
<tr>
<th></th>
<th>f_setosa</th>
<th>f_versicolor</th>
<th>f_virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Versicolor</td>
<td>0</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>Virginica</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>Precision</td>
<td>100%</td>
<td>96.0%</td>
<td>96.0%</td>
</tr>
<tr>
<td>Recall</td>
<td>100%</td>
<td>96.0%</td>
<td>96.0%</td>
</tr>
<tr>
<td>Average</td>
<td>97.3%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>97.3%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: The results of Kishore with SA measure

<table>
<thead>
<tr>
<th></th>
<th>f_setosa</th>
<th>f_versicolor</th>
<th>f_virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Versicolor</td>
<td>0</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>Virginica</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>Precision</td>
<td>100%</td>
<td>100%</td>
<td>89.3%</td>
</tr>
<tr>
<td>Recall</td>
<td>100%</td>
<td>88.0%</td>
<td>100%</td>
</tr>
<tr>
<td>Average</td>
<td>96.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>96.0%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

This paper presents an efficient algorithm to learn discriminating functions for classification based on genetic programming. The proposed approach includes an adaptive incremental learning strategy, a new fitness function and the adjusting method of Z-value measure. The experimental results show that the proposed method has high accuracy while two test beds, WDBC and Iris data sets, are evaluated. However, the attributes in the two data sets are all numerical. The categorical attribute seems to have no simple way to be applied for classification by discriminating functions. The future work on text classification based on discriminating functions is worth to be investigated.

REFERENCES


