Research Paper

DOI: 10.3969/j. issn. 1673-5188. 2017. S2. 009
http://kns.cnki.net/kcms/detail/34.1294.TN.20171208.1121.002.html, published online December 8, 2017

Random Forest Based Very Fast Decision Tree Algorithm for Data Stream

DONG Zhenjiang[1], LUO Shengmei[1], WEN Tao[1], ZHANG Fayang[2], and LI Lingjuan[2]
(1. ZTE Corporation, Nanjing 210012, China;
2. School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract

The Very Fast Decision Tree (VFDT) algorithm is a classification algorithm for data streams. When processing large amounts of data, VFDT requires less time than traditional decision tree algorithms. However, when training samples become fewer, the label values of VFDT leaf nodes will have more errors, and the classification ability of single VFDT decision tree is limited. The Random Forest algorithm is a combinational classifier with high prediction accuracy and noise-tolerant ability. It is constituted by multiple decision trees and can make up for the shortage of single decision tree. In this paper, in order to improve the classification accuracy on data streams, the Random Forest algorithm is integrated into the process of tree building of the VFDT algorithm, and a new Random Forest Based Very Fast Decision Tree algorithm named RFVFDT is designed. The RFVFDT algorithm adopts the decision tree building criterion of a Random Forest classifier, and improves Random Forest algorithm with sliding window to meet the unboundedness of data streams and avoid process delay and data loss. Experimental results of the classification of KDD CUP data sets show that the classification accuracy of RFVFDT algorithm is higher than that of VFDT. The less the samples are, the more obvious the advantage is. RFVFDT is fast when running in the multi-thread mode.

Keywords

data stream; data classification; Random Forest algorithm; VFDT algorithm

This work was supported by ZTE Industry-Academia-Research Cooperation Funds and National Natural Science Foundation of China under Grant Nos. 61302158 and 61571238.

1 Introduction

Classification on data streams refers to build a classification model for dynamic data streams.

A traditional classification algorithm classifies the existing data by a classifier, and does not need to construct any classifiers during the classification process. Compared with the supervised and semi supervised algorithms, its performance is greatly improved. However, the classification performance of a traditional classifier cannot meet the requirements of ever-changing data streams for real time and accuracy, so a series of data stream classification mining algorithms have emerged, such as Hoeffding Tree [1], [2], Very Fast Decision Tree (VFDT) [3], [4], Concept-Adapting Very Fast Decision Tree (CVFDT) [5], combination of classification and Iterative Dichotomiser 4 (ID4). Among them, VFDT is the most representative algorithm.

The classification accuracy of VFDT algorithm is generally similar to that of traditional decision tree algorithms, and when processing large amounts of data, VFDT needs less time. However, the label values of VFDT leaf nodes have more errors when training samples are fewer, therefore the classification ability of single VFDT decision tree is limited and the classification accuracy cannot be guaranteed.

The Random Forest algorithm is a combinational classifier with good classification performance [6]. It is constituted by multiple decision trees; it has high prediction accuracy and noise-tolerant ability, and can make up for the shortage of single decision tree.

In order to get higher classification accuracy on data streams, we integrate the Random Forest algorithm into the VFDT algorithm and design a Random Forest Based Very Fast Decision Tree algorithm, named RFVFDT. This algorithm introduces the criterion of building decision trees in the random forest classifier for the process of building the decision tree, and improves the Random Forest algorithm with sliding the time window to meet the unboundedness of data streams. In order to test the performance of RFVFDT algorithm, we have done experiments with KDD CUP data sets.

2 Related Work

2.1 VFDT Algorithm

A decision tree algorithm is the key method of data classification. There are many traditional decision tree algorithms such as ID3 and C4.5. These algorithms need to store the data and do batch processing, and they are not feasible for data streams with timeliness requirement.

The VFDT algorithm is proposed to construct the decision tree by training samples in real time with an acceptable cost, and its incremental feature can meet the timeliness requirement of data streams. It is realized by improving the Hoeffding

decision tree.

The Hoeffding tree is established by converting the leaf nodes into internal nodes continuously [1]. Each leaf node maintains statistical information about attributes; these statistics are used to calculate the information gain of the attribute. When a new sample coming, it traversals along the tree from top to bottom; each internal node in the tree does the division test; the sample belongs to different branches according to the different attribute values; eventually it reaches the leaf node of the tree and the statistics in the leaf node are updated. If the calculated statistics show that the result meets the Hoeffding boundary condition, the leaf node becomes an internal node, and new leaf nodes are generated based on the possible values of the internal node attribute.

In the process of building a decision tree, the VFDT algorithm uses information entropy or Gini index as the standard of choosing the splitting attribute [2], and uses Hoeffding inequality to determine whether to split the leaf node. The purpose of using Hoeffding inequality as node splitting condition is to determine the number of samples required for the leaf node changing into the internal node, so that the algorithm may use fewer samples to establish a decision tree with high accuracy. The related definitions are as follows:

Definition 1: Let t as a time stamp, xt expresses a data vector arriving at time t, and then data stream can be expressed as $\{...,xt-1,xt,xt-2,...\}$ [7].

Definition 2: For n independent observed values of a real-valued random variable r with range R, the mean value of them is $\bar{r}$. The Hoeffding bound ensures that the true value of r is at least $\bar{r}-\varepsilon$ with confidence $1-\delta$ [8], [9], i.e.

$$P(r \geqslant \bar{r}-\varepsilon)=1-\delta, \varepsilon=\sqrt{\frac{R^2 \ln(1/\delta)}{2n}}, \tag{1}$$

where r is the information gain, $R=\log_2 \#Classes$. The parameter Classes is the number of class attribute values.

Definition 3: A leaf node l stores statistics of sample set D. The expected information for classifying sample set D is

$$Info(D)=-\sum_{i=1}^{m} p_i \log_2(p_i), \tag{2}$$

where p_i is the probability of any one sample in sample set D belonging to the class of C_i, $p_i=\left|C_{i,D}\right|/\left|D\right|$, and m is the number of values of the class attribute. One leaf node has possible decision attribute A, and attribute A has v values. Then the expected information of classifying set D by attribute A is

$$Info_A(D)=-\sum_{j=1}^{v} \frac{\left|Dj\right|}{\left|D\right|} \times Info(Dj). \tag{3}$$

Therefore, the information gain of attribute A is $Gain(A)=Info(D)-Info_A(D)$ [10].

Definition 4: The active division coefficient τ is used to actively choose decision attributes and realize leaf node split when the values of several attributes' information gain G are almost equal. When $\Delta G<\varepsilon<\tau$ is met, an attribute with largest or the second largest information gain in ΔG is selected as the decision attribute of the leaf node.

Definition 5: One of the effective judgment conditions of node split is $(\overline{G_l(X_a)}-\overline{G_l(X_b)})>\varepsilon$ or $\varepsilon<\tau$, where $\overline{G_l(X)}$ is the information gain of attribute X in leaf node l, X_a is the attribute with the maximum information gain value, and X_b is the attribute with the second largest information gain value.

The process of classifying the samples using VFDT tree is as follows: 1) sending samples from the decision tree root to different branches according to the decision attribute and the corresponding sample values of the attribute; recurse the process until samples reach the leaf node; and 2) labeling the sample according to the class label of the leaf node. The label value of the leaf nodes is determined by the distribution of the class values of the training sample arriving at the leaf nodes. When the number of training samples is small, the number of training samples arriving at the leaf nodes is relatively small. Under this circumstance, the label of the leaf node may have more errors, and the classification ability of single VFDT decision tree is limited.

2.2 Random Forest

Random Forest [6] is a combinational classifier based on Boosting method [11] and with good classification performance. Like boosting method, Random Forest also consists of multiple decision trees. Final random forest classifier is formed by choosing a group of independent decision trees. The training set for each decision tree in the classifier is produced by randomly sampling with the bootstrap algorithm. The decision attribute of each leaf node in the decision tree is generated from a small set of attributes acquired in random.

The steps of the Random Forest algorithm are as follows: 1) The number of decision trees are set as N in the random forest, the bootstrap method is used to sample the static training set and produce N training sets, and the different training sets are independent and identically distributed; 2) For each training set, the decision tree is established independently. The splitting process of the leaf nodes includes: 1) selecting m (m << M) attributes randomly from the possible decision attributes of leaf node, where M is the number of attributes of the training sample; 2) selecting the decision attribute according to the minimum impurity principle, and realizing the growth of decision tree; and 3) marking the sample which needs to be classified according to the Random Forest classifier. Unlike the building of a traditional decision tree, the Random Forest tree does not need pruning.

The randomness of Random Forest is the randomness of the training samples of each decision tree and the randomness of the selection of the decision attributes of each leaf node in the decision tree. A lot of theoretical and experimental studies

Random Forest Based Very Fast Decision Tree Algorithm for Data Stream
DONG Zhenjiang, LUO Shengmei, WEN Tao1, ZHANG Fayang, and LI Lingjuan

have shown that the Random Forest algorithm has good prediction accuracy and good tolerance for outliers and noises, and is not easy to appear over-fitting.

3 Design of RFVFDT Algorithm

3.1 The Basic Idea

The RFVFDT algorithm has three parts: building and updating the classifier (or decision trees), classifying samples (or labeling samples) by the classifier, and evaluating the classifier.

In the process of building decision trees, the RFVFDT algorithm adopts the criterion of building decision trees in the random forest classifier. In addition, it adopts sliding-window to meet the unboundedness of data streams, and randomly obtains the sample from the sliding window based on the time granularity to guarantee the randomness of the samples. In the process of splitting leaf node, when a leaf node meets the condition $n_l \bmod n_{min} = 0$, the RFVFDT algorithm does not calculate the information gain of all possible attributes, but selects attributes randomly before calculating the information gain, and the decision attribute generates from the randomly selected attributes. Thus, the correlation of the decision tree in RFVFDT can be reduced greatly.

In the process of classifying samples, unlike the VFDT algorithm in which the label of a sample is only decided by the class label of the leaf node, the label of a sample in RFVFDT algorithm will be determined by the voting result from multiple decision trees.

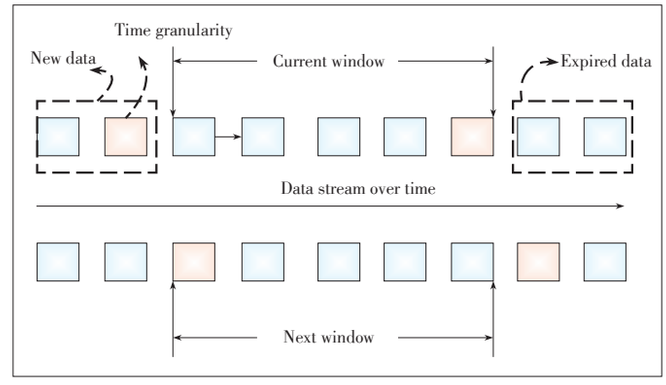
3.2 Building and Updating Classifier

In the RFVFDT algorithm, the training samples of each decision tree should be independent with each other. The Random Forest classifier generates N training sample sets by using the bootstrap sampling method, and forms the decision trees. Because the RFVFDT algorithm does classification on the data stream that is infinite, the bootstrap sampling method is not suitable for it. The sampling method we designed in RFVFDT is to cache training samples arriving in a period of time by the sliding window, dynamically update the samples in the sliding window as the time goes on, and randomly select each training sample of the decision tree from the sliding window. The sliding window based on time granularity is shown in **Fig. 1**.

Formally, R (N) indicates that there is N time granularities in the sliding window R, and 1 to N are continuous time blocks. In the sliding window, the data of the N continuous time granularities are always maintained. The data in a time granularity will be updated as a whole and the update frequency is lower.

In order to ensure the independence of training samples in each decision tree, the training sample of the RFVFDT classifier is randomly obtained from the current processing window.

The process of building decision trees of the RFVFDT classifier can be described as **Algorithm 1**.



▲**Figure 1. Sliding window based on time granularity.**

Algorithm 1. Building decision tree

Input: root, instance // The instance is the random training sample in sliding window
Output: Decision tree

1: Determine whether the current decision tree curTree exists. If not exists, initialize the tree, namely initialize a root and randomly select a small number of attributes as the attribute set for node splitting (the small number is sqrt (M), M is the number of the attributes of instance); otherwise do step 2
2: From the root node of the decision tree, the Instance traverses into different branches according to the node attribute values, until into the leaf node l, and the statistics of the leaf node are updated
3: if $n_l \bmod n_{min} = 0$ and not all instances belong to the same class then // n_l represents the number of samples in the leaf node l, n_{min} is a threshold for avoiding the excessive computation of the information gain of the node's attributes
4: Set temporary attribute space, randomly select m attributes from the M attributes of the training sample ($m << M$)
5: for the attribute in the random attribute set do
6: According to the statistical value of the leaf node l, calculate the information gain of the attribute, namely $\overline{G_l}$
7: end for
8: Find X_a, which is the attribute with highest value in $\overline{G_l}$
9: Find X_b, which is the attribute with second highest value in $\overline{G_l}$
10: Compute Hoeffding bound $\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n_l}}$
11: if $X_a \neq X_\phi$ and $((\overline{G_l(X_a)} - \overline{G_l(X_b)}) > \varepsilon$ or $\varepsilon < \tau)$ then // X_ϕ means null attribute
12: The leaf node converts into an internal node, and the attribute X_a is the decision attribute of the node.
13: Determine the number of branches of the split node according to the number of the values of attribute X_a
14: for all branches of the split do

15: For each branch add a new leaf node with related information
16: end for
17: end if
18: end if

The process of updating each decision tree in the classifier is as follows: The training sample starts from the root node of the decision tree, and enters a certain branch of the internal node according to its value of the attribute corresponding to the decision attribute of the internal nodes; the recursive loop is kept until it reaches the leaf node; after the training sample arrives at the leaf node l, the statistics are updated; when n_l mod n_{min} = 0, the algorithm would select the attributes randomly and then calculate the information gain of these selected attributes, and the decision attribute would also be generated from these attributes. The number of randomly selected attributes is noted as m, and m is generally set as sqrt (M) or sqrt (M)/2, and M is the total number of attributes of the training sample. When the conditions of the node splitting, i.e. $X_a \neq X_\phi$ and $((\overline{G_l(X_a)} - \overline{G_l(X_b)}) > \varepsilon or \varepsilon < \tau)$, are satisfied, the leaf node splits.

3.3 Classifying Samples

For each RFVFDT decision tree in the classifier, the samples to be classified go from the root into different branches based on the value of decision attribute of each internal node, and reach the leaf node after top-down traversal. The label of the sample will be decided by the label of the leaf node.

In RFVFDT, the same sample is marked by all the decision trees in the random forest classifier, and the final label is decided by voting. In other words, RFVFDT labels each sample to be classified in every decision tree of the random forest classifier, and then the label of the sample is decided by the class label which is in the majority. That is the voting process of the classifier. To a certain extent, voting by the classifier can avoid the poor classification performance of single decision tree.

Algorithm 2 shows the process of classifying samples by the RFVFDT classifier.

Algorithm 2. Classifying samples by RFVFDT classifier

Input: RFVFDT classifier, unInstance // unInstance is the sample to be classified
Output: The label of unInstance
1: for each decision tree in the RFVFDT classifier do
2: Label unInstance by a decision tree rfvfdt$_i$ with label$_i$
3: Make real-time update on the map (label$_i$, integer), where label$_i$ is the label made by rfvfdt$_i$ and the integer is the number of label$_i$ in the classification results of each decision tree
4: end for
5: Use the label with the largest corresponding integer in the map to label unInstance

3.4 Evaluating Classifier

Evaluating the RFVFDT classifier is implemented by comparing the experimental results of each labeled sample enInstance in the test sample set with the actual label value of the sample to get the overall evaluation value. The process of evaluating RFVFDT classifier is described in **Algorithm 3**.

Algorithm 3. Evaluating RFVFDT classifier

Input: RFVFDT classifier, the test sample set enInstances
Output: Evaluation results
1: Initialize statistics cou = 0
2: for each labeled enInstance in the test sample set enInstances do
3: Call the method of classifying the samples, and return the experimental label value
4: Compare the label value with the actual label of the enInstance, and do cou+1 if the values are the same
5: end for
6: return 100.0*cou/(the number of samples in EnInstances)

4 Performance Analysis of RFVFDT Algorithm

In order to test the performance of RFVFDT algorithm, we conducted several experiments to compare the RFVFDT algorithm with VFDT.

4.1 Experimental Environment and Data Set

The experimental environment is java/JDK 1.7, eclipse 4.4.2, win7 Home Basic 32 bits, and PC with 2.13 GHz, 4 GB. The data sets of KDD CUP are used in the experiments.

We tested the classification performance of the RFVFDT classifier with training samples increased. We used two training sets separately to avoid the chanciness of the experimental results on a single training sample set. The data sets are shown in **Table 1**. In order to simulate the data streams, the data read do not be put back until all the training sample studies are completed. The classification performance is determined by the average of the results from multiple tests.

4.2 Accuracy of RFVFDT Classifier with Training Samples Increased

Fig. 2 shows the comparison results of the classification ac-

▼**Table 1. Data set used in the experiments**

Data set	NA	NCA	NTA_1	NTA_2
Nursery	9	5	380,000	10,000
Connection	42	3	380,000	10,000

NA: the number of attributes
NCA: the number of the values of class attributes
NTA_1: the number of training samples
NTA_2: the number of test samples

curacy of the VFDT decision tree and RFVFDT classifier with different data sets and increasing training samples. The parameters of the VFDT algorithm are set as: $\delta = 10^{-4}$, $\tau = 5\%$, $n_{min} = 200$, and there is no repeated scan for the samples. The parameters of decision trees in the RFVFDT classifier are consistent with the VFDT algorithm, and the number of decision trees in RFVFDT classifier is set to 30.

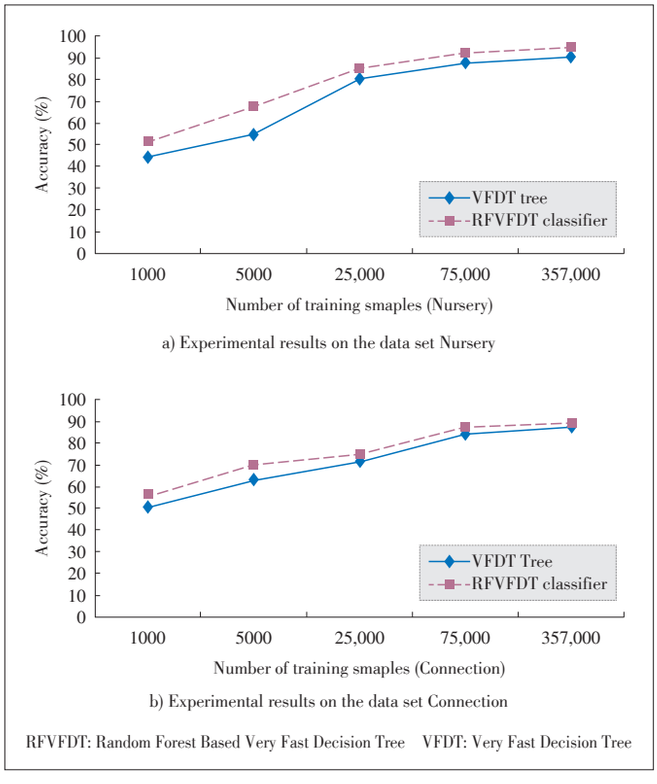
From the results in Fig. 2, it is concluded that the classification performance of each classifier is improved with the increasing of the training sample, that the classification accuracy of the RFVFDT classifier proposed in this paper is higher than that of the VFDT algorithm, and that the accuracy of the RFVFDT classifier is more higher than that of the VFDT algorithm under the condition of the training sample is relatively small.

4.3 Processing Efficiency of RFVFDT Classifier

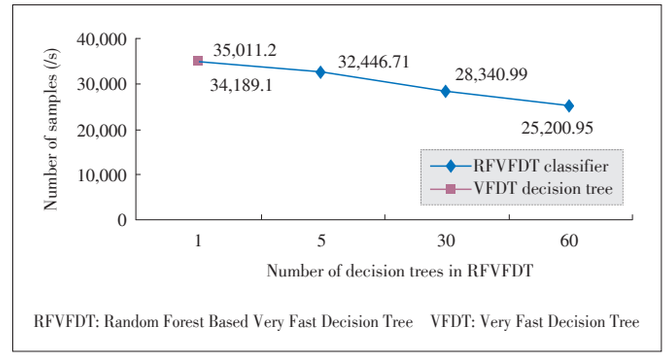
We use the number of samples processed per second to express the processing efficiency.

Using the data set Nursery, we established a VFDT decision tree and a number of RFVFDT classifiers with different numbers of decision trees, and tested the processing efficiency of each classifier with the same classification samples. **Fig. 3** shows the results and the algorithms are running in a single thread serial mode.

As can be seen from Fig. 3, the processing efficiency of the



▲**Figure 2. The classification accuracy with the size of the training sample set increased.**



▲**Figure 3. The processing efficiency with different numbers of decision trees.**

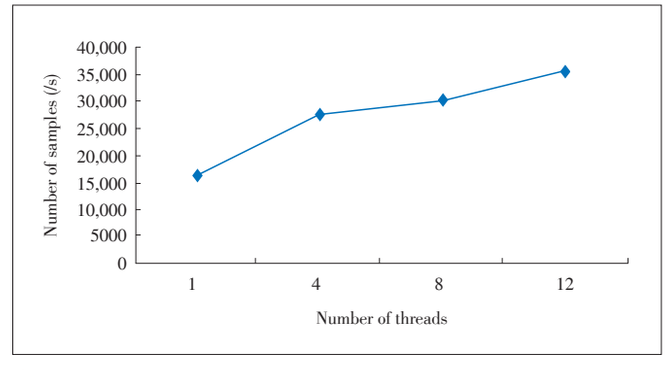
RFVFDT classifier declines with the number of decision trees in the RFVFDT classifier increased. The reason is that in order to ensure the classification accuracy, RFVFDT maintains a certain number of decision trees in the classifier. The sample should be marked by each decision tree in the RFVFDT classifier, and the label is voted by the label from each decision tree.

Still using the data set Nursery, we run the RFVFDT algorithm in a multi-thread parallel mode, and the number of decision trees in the RFVFDT classifier is set to 30. The processing efficiency testing results are shown in **Fig. 4**.

As can be seen from Fig. 4, the processing efficiency of RFVFDT increases with the number of threads increased, which means if we run the RFVFDT algorithm under the multi-thread environment such as a cluster, it can keep higher classification accuracy and a fast speed as well.

5 Conclusions

In this paper, based on the Random Forest algorithm and the data stream classification algorithm VFDT, we designed a Random Forest Based Very Fast Decision Tree algorithm, named RFVFDT. The analysis and the experimental results of doing classification on KDD CUP 99 data sets have shown that the classification accuracy of the RFVFDT algorithm is higher than that of VFDT, and the RFVFDT algorithm can keep fast



▲**Figure 4. The processing efficiency of RFVFDT with different numbers of threads.**

and also achieve higher classification when parallelly running with multiple threads.

References

[1] M. Guo, "Research and design of real-time traffic classification for high-speed network," MS thesis, Dept. of Computer application technology, Beijing University of Posts and Telecommunications, Beijing, China, 2010.
[2] A. Bifet and G. De Francisci Morales, "Big data stream learning with samoa," in *IEEE International Conference on Data Mining Workshop*, Shenzhen, China, Dec.2014, pp. 1199—1202. doi: 10.1109/ICDMW.2014.24.
[3] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, 2000, pp. 71—80. doi: 10.1145/347090.347107.
[4] J. Gama, R, Rocha, and P. Medas, "Accurate decision trees for mining high speed data stream," in *Proc. Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, 2003, pp. 523—528. doi:0.1145/956750.956813.
[5] B. Raahemi, W. Zhong, and J. Liu, "Peer-to-Peer Traffic Identification by Mining IP Layer Data stream Using Concept-adapting Very Fast Decision Tree," in *20th IEEE International Conference on Tools with Artificial Intelligence*, Dayton, USA, Nov. 2008, pp. 525—532.doi: 10.1109/ICTAI.2008.12.
[6] L. Breiman, "Random forests," *Machine Learning*, vol. 44, no. 1, pp. 5—32, Oct.2001. doi: 10.1023/A:1010933404324.
[7] T. Wang, Z. Li, X. Hu, Y. Yan, H. Chen "An incremental fuzzy decision tree classification method for data stream mining based on threaded binary search trees," *Chinese Journal of Computers*, vol. 30, no. 8, pp. 1244—1250, Aug. 2007. doi:10.3321/j.issn:0254-4164.2007.08.005.
[8] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Machine Learning*, vol. 2, no. 4, pp. 285-318, Apri. 1988. doi:10.1007/BF00116827.
[9] O. Maron, "Hoeffding Races--model selection for MRI classification," MS thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA, 1994.
[10] L. Jiang, Z. Cai, and Z. Liu, "An algorithm of classification rules mining based on information gain," *Journal of Central South University of Technology*, vol. 34, no. 2, pp.69—71, July 2003.doi:10.3969/j.issn.1672-7207.2003.z1.020.
[11] A. Wang, G. Wan, and Z. Chen, "Incremental extreme random forest classifier for online learning," *Journal of Software*, vol. 22, no. 9, pp. 2059—2074, Sept. 2011. doi: 10.3724/SP.J.1001.2011.03.827.

Manuscript received: 2016-12-02

Biographies

DONG Zhenjiang (dong.zhenjiang@zte.com.cn) received his M.S. degree in telecommunication and electronics from Harbin Instituted of Technology in 1996. He is the deputy head of the Service Institute of ZTE Corporation. His research interests include cloud computing and the mobile Internet.

LUO Shengmei (luo.shengmei@zte.com.cn) received his M.S. degree in telecommunication and electronics from Harbin Instituted of Technology in 1996. He is now a chief architect at ZTE Corporation. His research interests include big data, cloud computing and network storage.

WEN Tao (wen.tao1@zte.com.cn) received his M.S. degree in computer science from Nanjing University of Aeronautics and Astronautics, China. He is a senior pre-research engineer at the Cloud Computing & IT Institute of ZTE Corporation. His research interests include cloud computing and big data technologies.

ZHANG Fayang (13041105@njupt.edu.cn) is pursuing his master degree at School of Computer, Nanjing University of Posts and Telecommunications, China. His research interests include cloud computing, data mining, and big data technologies.

LI Lingjuan (lilj@njupt.edu.cn) is a full professor of School of Computer, Nanjing University of Posts and Telecommunications, China. She received her Ph.D. degree in computer application technology from Soochow University, China. Her research interests include cloud computing, data mining, information security, and big data technologies.