Detection of Fraud Use of Credit Card by Extended VFDT

Tatsuya Minegishi[1], Ayahiko Niimi[2]
[1]*Graduate School of Systems Information Science,* [2]*Faculty of Systems Information Science*
Future University Hakodate
g2109043, niimi{ @fun.ac.jp}

Abstract

Global society has experienced a flood of various types of data as well as a growing desire to discover and use this information effectively. Moreover, this data is changing in increasingly huge and complex ways. In particular, for data that is generated intermittently and at different intervals, attention has been focused on data streams that use sensor-network and stream mining technologies to discover useful information. In this paper, we focus on classification learning, which is an analytical method of stream mining. We are concerned with a decision tree learning called Very Fast Decision Tree learner (VFDT), which regards real data as a data stream. We analyze credit card transaction data as data stream and detect fraud use. In recent years, people with credit card are increasing. However, it also increases the damage of fraud use accordingly. Therefore, the detection of fraud use by data stream mining is demanded. However, for some data, such as credit card transaction data, contains extremely different rate of classes. Therefore, we propose and implement new statistical criteria to be used in a node-construction algorithm that implements VFDT. We also evaluate whether this method can be supported in imbalanced distribution data streams.

1. Introduction

Recent developments in information processing techniques have enabled us to collect and accumulate massive amounts of data. The need for discovering and utilizing the useful information in this data is growing. Because of this, data mining, which is a technology used to collect data to discover useful information, has become a focus of attention. However, with the spread of the Internet and the development of sensor techniques, this data is constantly evolving into more complex shapes on a large scale, and the increasing data must be handled on a real-time basis. New knowledge-stream-mining techniques are required to process such large-scale data that arrives intermittently and at different intervals as data-stream flows. Stream mining uses various analytical methods; in particular, classification learning is gaining much attention. Many classification learning methods have been proposed, of which the decision tree learning method is commonly used, because it is fast and the description of classifiers that it derives is easily understood. One of the data streams that supports the decision tree learning method is called the Very Fast Decision Tree (VFDT)[1]. As data arrives, this data stream grows gradually while the data is classified. The credit card transaction data is data stream. Therefore it is possible to detect fraud use to classify transaction data using VFDT. However, among the various data types, there is some data, such as the credit card transaction data discussed in this paper whose characteristics are extremely different. When such data is used in a data stream, some problems are capable of causing the accuracy of VFDT to be decreased[2,3].

In this paper, we propose a node-construction algorithm that can apply to imbalanced distribution data streams. We also implement and evaluate the criteria for constructing nodes.

This paper is organized as follows. First, in section 2, we explain the VFDT. In section 3, we describe our proposed method, which consists of a VFDT construction from imbalanced distribution data streams. In section 4, we verify the viability of the proposed methods in experiments. In sections 5 and 6, we describe and consider the experimental result. In the final section, we conclude and discuss our future works.

2. Related works

Classification is one of the most common tasks in data mining. The main classification methods that currently exist include decision trees, neural networks, logistic regression, nearest neighbors, and support vector machines.

Decision trees are recognized as very powerful and attractive classification tools, mainly because they produce easily interpretable and well-organized results and are, in general, computationally efficient and capable of dealing with noisy data. Decision tree techniques build classification or prediction models

based on the recursive partitioning of data, which begins with the entire body of data then splits the data into two or more subsets based on the values of one or more attributes, and then repeatedly splits each subset into finer subsets until the stopping criteria are met [4].

Typical decision tree learning methods include ID3 and C4.5[5]; however, they cannot correspond to data streams. The VFDT has extended these decision tree learning methods to correspond to data streams. In addition, there are the CVFDT[6], $CVFDT_{NBC}$ [7], and UFFT[8] methods, which are considered concept drift methods, which are useful when the properties of the data stream change over time.

In this paper, we don't refer to concept drift and pick up VFDT.

2.1. VFDT

A decision tree construction such as C4.5, which first receives all examples as input, is called an offline type decision tree. However, this method cannot start constructing until all the examples are available, and it also needs to access them randomly. Therefore, it cannot be applied to data streams.

On the other hand, a decision tree construction in which new examples arrive in sequence at short intervals in a data stream and huge numbers of examples accumulate is called an online type decision tree. A representative example is the Very Fast Decision Tree (VFDT) learner.

The VFDT does not accumulate the examples in main memory, because it can gradually grow without waiting for the arrival of all the examples. The construction algorithm of the VFDT accumulates only the classes of examples and the contemporaneous occurrence frequency of attribute values in each node to decrease the consumption of memory and processing time, instead of accumulating examples in a decision tree. The VFDT gradually grows as examples are received to create leaf nodes that grow into branches from only the root node. When it creates new nodes, it grows the decision tree, accumulating frequency information in the previous node and measuring whether the new nodes fulfill the statistical criteria.

The statistical criterion called the Hoeffding bound[9] is used by the VFDT. The examples accumulated in leaf nodes are only a part of all the available examples. Therefore, it is possible that they include errors. However, the set of examples that arrive at each leaf node can be regarded as perfect data sets in an offline type decision tree, which can consider infinitely-long data streams produced stochastically based on stationary distribution.

Consider a real-valued random variable *r* having a range *R* and conduct *n* independent observations of this variable. After computing their mean $\bar{r}$, the Hoeffding bound guarantees that the true means of variable r is greater than $\bar{r} - \varepsilon$ with a probability of $1 - \delta$. Here, ε is defined as follows:

$$\varepsilon = \sqrt{\frac{R^2 \ln(1/\varepsilon)}{2n}} \qquad (1)$$

If the difference between the best standard level at one leaf and the next standard level is greater than ε, then it creates additional branches from the leaf node.

Using the Hoeffding bound, if $\overline{\Delta G()} - \overline{G(X_a)} - \overline{G(X_b)} > \varepsilon$, then splitting node by attribute X_a with probability $1 - \delta$ is true. Here, $G()$ is an information-gain function, where X_a is the attribute that creates the largest information gain and X_b is the attribute that creates the second-largest information gain.

2.2. VFDT construction from data stream

In the current research, previous study constructed the VFDT, which is a decision tree learning method that corresponds to the data stream[10].

Here, we constructed the VFDT to consider credit card transaction data as a data stream.

As described in section 1, credit card transaction data is extremely different from the rate of classes of data in classification; however, we constructed the VFDT without adding the change to the construction algorithm of the VFDT. To verify whether the VFDT changes by data sampling, we constructed 10 VFDTs using 10 data sets when constructing the VFDT using the credit card transaction data. Because the best-grown decision tree is 10% in constructing decision trees by three types of fraud-use-rate experiments in the construction of offline type decision trees in existing research[11], here, we set the fraud-use rate to 10%. In offline type decision trees that are constructed by C4.5, the fraud use rate setup is as follows:

(a) 0.02% is the actual fraud-use rate
(b) 0.5% is the sampling rate of data provided
(c) 10% is set up in the experiment.

(a) becomes a decision tree divided into two leaf nodes by the root node. (b) becomes a decision tree that has 101 nodes, including 51 leaves.

Both (a) and (b) have more than 99% accuracy, but both the fraud use rates 0.02% and 0.5% are very low to actually classify almost all the fraud data. Therefore, we used a data set of 10% fraud-use rate to construct the VFDT.

To evaluate by 10-fold cross-validation in all VFDTs and retrieve their accuracy and size, we calculated the average of the results of 10 decision trees. Therefore, we actually constructed 100 VFDTs. The accuracy of the VFDT became

92.157%, and the size of the VFDT became 91. The result of the VFDT is independent of data sampling because the variance of accuracy became 0.290.

3. Extended VFDT for imbalanced distribution data stream

The Hoeffding bound, which is the node construction criterion described in 2.1, assumes that the data distribution of the data stream uses Gaussian distribution[12]. However, the credit card transaction data described in 4.1.1 contains classes of examples that include data streams that do not follow Gaussian distribution. In this case, the accuracy of the constructed VFDT is high but actual classification accuracy of one class is almost ignored.

Therefore, we propose the construction of a VFDT that can correspond to imbalanced distribution data streams, in order to improve the calculation of the Hoeffding bound, which is the node-construction criterion of the VFDT. We weight the entropy of $G(X_a)$ and $G(X_b)$ using the calculation of information gain $\overline{\Delta G()} = \overline{G(X_a)} - \overline{G(X_b)} > \varepsilon$ by judging when it grows new branches from leaf nodes. In this paper, we define two classes.

The calculation of entropy using ID3 and C4.5 defines that $freq(C_i, S)$ to set of examples S is the number of examples that are in class C_i in S, the number of examples including set S is $|S|$ and an example selected randomly from S is in class C_i.

Therefore, the average entropy *info(S)* is as follows:

$$info(S) = -\sum_{i=1}^{2} \frac{freq(C_i,S)}{|S|} log_2\left(\frac{freq(C_i,S)}{|S|}\right) \qquad (2)$$

Here, we weight the entropy of each class and assume a sum when calculating $\overline{G(X)}$. The weight has the range of $0 \leq \omega \leq 1$ and is a class of fraud use. Therefore, if the class of fraud use is C_1 and the class of normal use is C_2, then:

$$info(S) = -\omega \frac{freq(C_1,S)}{|S|} log_2\left(\frac{freq(C_1,S)}{|S|}\right) - (1-\omega) \frac{freq(C_2,S)}{|S|} log_2\left(\frac{freq(C_2,S)}{|S|}\right) \qquad (3)$$

Additionally, the VFDT does not support numeric data streams, because it is an algorithm with a discrete data stream. However, the programs of the VFDT released in VFML[13], which is the tool used to construct the VFDT, include improvements for handling numeric attributes. Specifically, entropy-based discretization[12] has been adopted. When the VFML discretizes numeric attributes to two intervals by attribute value, which increases the maximum information gain of each numeric attribute, we also similarly weight the calculation of information gain.

However, only the information gain of the left part is weighted when it compares the information gain after weighting the Hoeffding bound as $\overline{\Delta G()} = \overline{G(X_a)} - \overline{G(X_b)} > \varepsilon$. Therefore, if it compares directly the Hoeffding bound ε of the right part, then it compares weighted to nonweighted. Hence, we multiply the information gain of the right part by the average 0.5 of ω and $1-\omega$ to two classes, and we balance both sides.

4. Experiments

In this section, we describe some experiments to verify the effectiveness of our proposal method and its evaluations.

4.1. Experimental data and tools

Here, we describe data and tools used in experiment.

4.1.1. Credit card transaction data

In this paper, we regard credit card transaction data as a data stream and conduct experiments using it. In actual credit card transactions, the data is complex, constantly changing and arrives online continuously as follows:

(i) Approximately one million transactions arrive per day.
(ii) Each transaction takes less than one second.
(iii) Approximately one hundred transactions arrive per second at peak time.
(iv) Transactions arrive 24 hours per day, every day, and continue to arrive forever.

Therefore, credit card transaction data can be precisely called a data stream. However, even if we use data mining for such data, an operator can generally accommodate monitoring around only 2,000 transactions per day. Therefore, we have to detect suspicious transaction data effectively under the rigid conditions of 0.02% of the total number of transactions. In addition, there is the issue that people detect extremely low fraud use from massive amounts of transaction data, because real fraud use occurs at an extremely low rate, that is, from 0.02% to 0.05% of all of the transaction data.

The data that we use in this paper describes transaction data in CSV format in time order and the data exists as attributes. Credit card transaction data has 124 attributes: 84 are called transactional data, which includes an attribute to discriminate whether the data is fraud use, and the others are called behavioral data, which are calculated by a user's usage. The file size is approximately 700 MB per month. As mentioned earlier, the fraud-use rate is from 0.02% to 0.05% before, and this data is re-sampled to about 0.5%.

- The number of attributes of data

- 57 transactional attributes and 42 behavioral attributes

- Sampling rate of fraud use

- Fraud : Normal = 1 : 9

Usually, the provided data has around 120 attributes; however, we exclude some attributes that are irrelevant for the construction of the decision tree, which has low relevance for fraud-use models. We also use approximately 50,000 data items.

4.1.2. UCI data

We performed experiments for benchmark testing using the spambase data set of the UCI data set[14]. The reason being that it has two classes defined in 3, all attributes are numeric like credit card transaction data, and it did not become the VFDT, which has only the root node without changing algorithm. Following are the contents of the spambase data:

- Number of attributes of data : 57
- Class : 1(Spam class), 0(non-spam class).
- Number of data : spam class : 1,813, non-spam class : 2,788

4.1.3. Tools

We constructed these experiments using VFML, which is implementation code for the machine learning of a data stream. Next, we constructed an offline type decision tree using the J48 algorithm based on C4.5 and implemented in data-mining tool software called Weka[15]. Then, we compare these results with the proposed VFDT.

4.2. Experimental methodology

We constructed the following two VFDTs using the two sets of data described in 4.1.

(i) A VFDT using the VFDT construction algorithm implemented using the VFML without change.

(ii) A VFDT whose entropy uses the calculation of the information gain to grow new branches from leaf nodes and in which discretized numeric attributes are weighted.

As we described in 3, We weighted credit card transaction data to focus on the distribution of its data classes. As described in 4.1.1, we re-sampled the ratio of the normal and fraud as 9:1. We set $\omega = 0.9$ based on the accuracy of the VFDT, considering the data distribution. We also performed experiments for the case of $\omega = 0.1,0.5,0.99,0.999$. This is in case the weight to fraud use is small, the weight conforms to normal use and the weight approximates 1. We excluded $\omega = 0.0,1.0$ because only the entropy of the one class is calculated by these weights. (The entropy of the other class becomes 0.)

We also weighted one class as ω. The range is $0 \leq \omega \leq 1$. In the experiments using the spambase data set, we set $\omega = 0.1,0.2,\ldots,0.8,0.9$ at 0.1 intervals. We excluded $\omega = 0.0,1.0$ for similar reasons as in the credit card transaction data.

In the case of both data sets, $\omega = 0.5$ is the same as the existing method without weighting, because the operation to compare the information gain to the Hoeffding bound is applied. In both experiments, the pruning of the VFDT is set to $pruning\ confidence = 25\%$ as the default value.

5. Experimental results

Here, we show the results of each experiment.

5.1. Results of credit card transaction data

Table.1 lists the accuracy, the size, the runtime and the number of fraud rules of the VFDT constructed in (i) and (ii) of 4.2 using credit card transaction data. Table.2 lists the result of the confusion matrix given from the existing method corresponding to $\omega = 0.5$. The result of the confusion matrix given $\omega = 0.1,0.9,0.99,0.999$ is listed in order from Table.3 to Table.6. The result of Table.1 is that of the result of performing 10-fold cross-validation implemented in VFML. VFML calculates the error rate, the number of all the nodes of the VFDT as the size and the runtime. Using 10-fold cross validation, we actually calculate the average of 10 VFDTs in each weight.

Table 1. The accuracy, the tree size, the runtime and the number of fraud rules

	Accuracy	Tree size	Runtime	Fraud rules
$\omega = 0.5$	90.851	91.000	5.907	3
$\omega = 0.1$	71.188	27.400	4.289	3
$\omega = 0.9$	92.325	106.600	6.722	5
$\omega = 0.99$	90.881	99.400	6.632	3
$\omega = 0.999$	89.879	90.800	6.433	3

Table 2. Confusion Matrix (Existing method)

$\omega = 0.5$		Actual classes	
		0(Normal)	1(Fraud)
Leaf classes	0(Normal)	40,825	2,174
	1(Fraud)	1,494	2,598

Table 3. Confusion Matrix (Proposal method)

$\omega = 0.1$		Actual classes	
		0(Normal)	1(Fraud)
Leaf classes	0(Normal)	32,027	1,550
	1(Fraud)	10,292	3,222

Table 4. Confusion Matrix (Proposal method)

$\omega = 0.9$		Actual classes	
		0(Normal)	1(Fraud)
Leaf classes	0(Normal)	40,174	1,982
	1(Fraud)	2,145	2,790

Table 5. Confusion Matrix (Proposal method)

$\omega = 0.99$		Actual classes	
		0(Normal)	1(Fraud)
Leaf classes	0(Normal)	41,449	3,555
	1(Fraud)	870	1,217

Table 6. Confusion Matrix (Proposal method)

$\omega = 0.999$		Actual classes	
		0(Normal)	1(Fraud)
Leaf classes	0(Normal)	42,252	4,669
	1(Fraud)	67	103

5.2. Results of spambase data set

Table.7 lists the accuracy, the size and the runtime of the VFDT constructed in (i) and (ii) of 4.2 using the spambase data set. Table.8 lists the result of the confusion matrix given from the existing method corresponding to $\omega = 0.5$. Table.9 lists the result of confusion matrix weighted $\omega = 0.1,0.2,0.3,0.4$. The result of the confusion matrix given $\omega = 0.6,0.7,0.8,0.9$ is listed in order from table.10 to Table.13. The reason that we coordinate the results from $\omega = 0.1$ to $\omega = 0.4$ in Table.9 is that all the results conform, because the classification rules of weighted class 1 (spam class) of the VFDT are all the same. The result of Table.7 is that of the performing 10-fold cross-validation. The accuracy in Table.7 is that we subtract the error rate from 100%. Using 10-fold cross-validation, we actually calculate the average of 10 VFDTs in each weight.

Table 7. The accuracy, the size and the runtime

	Accuracy(%)	Tree size	Runtime
$\omega = 0.1$	78.131	3.000	0.281
$\omega = 0.2$	78.941	3.800	0.257
$\omega = 0.3$	79.101	4.200	0.254
$\omega = 0.4$	80.500	7.200	0.293
$\omega = 0.5$	80.296	9.400	0.341
$\omega = 0.6$	76.564	17.000	0.474
$\omega = 0.7$	75.879	16.000	0.466
$\omega = 0.8$	70.116	11.200	0.396
$\omega = 0.9$	69.149	11.000	0.393

Table 8. Confusion Matrix(Existing method)

$\omega = 0.5$		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	2,751	1,092
	1(spam)	73	721

Table 9. Confusion Matrix(Proposal method)

$\omega = 0.1,0.2,0.3,0.4$		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	2,198	193
	1(spam)	590	1,620

Table 10. Confusion Matrix(Proposal method)

$\omega = 0.6$		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	2,463	770
	1(spam)	325	1,043

Table 11. Confusion Matrix(Proposal method)

$\omega = 0.7$		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	2,389	480
	1(spam)	399	1,333

Table 12. Confusion Matrix(Proposal method)

$\omega = 0.8$		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	2,542	1,205
	1(spam)	246	608

Table 13. Confusion Matrix(Proposal method)

$\omega = 0.9$		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	2,542	1,205
	1(spam)	246	608

6. Considerations

Here we examine each of the experiments.

6.1. Credit card transaction data

As described in Table.1, the accuracy of the VFDT that was constructed using credit card transaction data became the highest (92.325%) at $\omega = 0.9$, and it became the lowest (71.188%) at $\omega = 0.1$. From Table.2 to Table.6 of the confusion matrix, Fig.1 shows the recall to fraud use class.

When $\omega = 0.1$, the fraud use is detected the most. When $\omega = 0.9$, the accuracy of the VFDT is second. However, when $\omega = 0.1$, the number of normal use

classified to leaf nodes which is normal use are extremely fewer than another results of weight. Therefore, we think that the accuracy of the VFDT is low. Additionally, in the case of approximating weight to fraud use by 1, such as $\omega = 0.999$, the recall of fraud use became very low. Therefore, it can be said that the accuracy of the VFDT itself consists almost entirely of the classification of normal use.

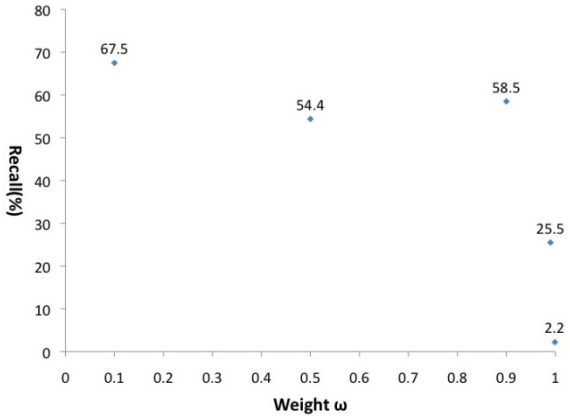

Figure 1. The recall of the fraud use

Moreover, we compare the size and the number of the fraud use rules of the VFDT with each weight value; they are stationary with the exception that the size became 27.400 with $\omega = 0.1$. With the exception that the number of the fraud rules is five, in the case that the size of the VFDT became the largest with $\omega = 0.9$, the number of the fraud use rules is three.

To compare the results of the credit card transaction data, we constructed an offline type decision tree using the C4.5 algorithm. The accuracy of the VFDT itself became 95.459%. The offline type decision tree constructed using the C4.5 algorithm constructs to input all of the data first. Therefore, its accuracy was usually higher than the accuracy of the VFDT, which is grown using partial data from all of the data. Table.14 lists the confusion matrix of this tree. As in the case of the VFDT, we calculated the recall to fraud use, which became 79.742%. For the VFDT constructed using credit card transaction data, the recall to weighted fraud use decreased as compared with offline. As described in Table.1, the runtime for VFDT in case of changing the value ω are no significant change. However VFDT can construct in 1/20 the time it took for constructing by C4.5. This is that VFDT constructs gradually to divide data. By contrast, C4.5 constructs to input all of the data first.

Table 14. Confusion Matrix(Fraud use by C4.5)

		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	41,422	897
	1(spam)	1,241	3,531

6.2. Spambase data set

As described in table.7, the accuracy of the VFDT, which was constructed using the spambase data set, became the highest (80.500%) at $\omega = 0.4$, and it became the lowest (69.149%) at $\omega = 0.9$. From Table.8 to Table.13 of the confusion matrix, Fig.2 shows the recall to weighted class 1 (spam class).

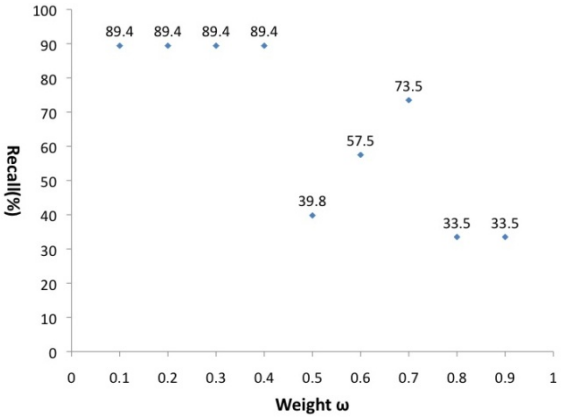

Figure 2. The recall of the spam class

When $\omega = 0.1,0.2,0.3,0.4$, it became the highest and the VFDT best classified the spam class described in Fig.2. In the case of $\omega = 0.1,0.2,0.3,0.4$, the shape of the VFDT is exactly the same and there is only one rule of the spam class. Therefore, the confusion matrix is the same.

As in the case of the credit card transaction data, we constructed an offline type decision tree using the C4.5 algorithm. The accuracy of the decision tree itself became 92.980%. Table.15 lists the confusion matrix of this tree.

Table 15. Confusion Matrix(spambase by C4.5)

		Actual classes	
		0(non-spam)	1(spam)
Leaf classes	0(non-spam)	41,422	897
	1(spam)	1,241	3,531

As in the case of the VFDT, we calculated the recall to the spam class from the confusion matrix. It became 90.789%. The accuracy of the decision tree itself is superior to the accuracy of the VFDT, but the recall to class 1 of the VFDT that is weighted decreased only 1% as compared with offline. As

described in Table.7, the runtime for VFDT in case of changing the value ω are no significant change. However VFDT can construct in 1/3 the time it took for constructing by C4.5. Consequently, to construct the VFDT to weight, we can improve the classification accuracy of the weighted class as much as the offline type decision tree.

6.3. Comparison of the results

We weighted the class whose ratio of distribution is small. Therefore, we considered that the classification accuracy of the class would increase and the accuracy of the VFDT itself would increase as well. However, in the case of the spambase data set, both the accuracy of the VFDT itself and the recall to class 1 were high when the weighting was small. In the case of the credit card transaction data, when the weighting was large, the accuracy of the VFDT itself improved. However, the recall to fraud use was high when the weighting was small. However, in the credit card transaction data, the recall of the normal use extremely decreased when the weight was small. Therefore, we proved that the accuracy of the VFDT itself decreased.

For both cases, in terms of the weight class, we can improve the accuracy of the VFDT itself and the recall. Especially, we can improve the accuracy more than C4.5 algorithm's one.

However, in these experiments, we could not specify how to weight to improve the accuracy of the VFDT. Because the ratio of the distribution of the class of the spambase data set and the credit card transaction data is extremely different, the accuracy does not improve to weight in the case of the spambase.

7. Conclusion and future works

In this paper, we used credit card transaction data as an imbalanced distribution data stream. We proposed and implemented a new statistical criterion for a node-construction algorithm, and verified its viability. As a new statistical criterion for a node-construction algorithm, we weighted the class of the entropy by comparing the Hoeffding bound to the information gain used in splitting the nodes of the existing algorithm. We think that the accuracy will improve if we weight the data of the small ratio of the class with a large weight. We verify that we can apply the proposed method to not only imbalanced distribution data streams such as credit card transaction data, but also to usual data streams. For this, we considered the results of the VFDT, which is constructed using the UCI data set.

As a result, we can improve the accuracy and the recall to weight one class with both data sets. Especially, depending on the weight values, the accuracy of the VFDT is the same as the offline type decision tree constructed using a C4.5 algorithm.

However, using the credit card transaction data and the credit card transaction data in these experiments, the accuracy of the VFDT itself and the recall to weighted class have variability, and the result do not conform with the same weight. For this reason, from a usual data stream and a data stream whose ratio of distribution of data is extremely different, we cannot find the best means for weighting in these experiments.

In future works, we will consider how to decide the weight without pre-experiments.

8. Acknowledgment

Thanks to the data, various pieces of advice, and guidance for the experiment given by relevant persons in INTELLIGENT WAVE INC.

9. References

[1] P. Domingos and G. Domingos. Mining High-Speed Data Streams. Proceedings of the ACM Sixth International Conference on Knowledge Discovery and Data Mining, ACM Press: 71–80, 2000.

[2] Chris Drummond, Robert C. Holte. Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria. Proceedings of Seventeenth International Conference on Machine Learning, 239–246, 2000.

[3] David A. Cieslak and Nitesh V. Chawla. Learning Decision Trees for Unbalanced Data. Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases, 241–256, 2008.

[4] H. Hu, Y. Chen and K. Tang. A Dynamic Discretization Approach for Constructing Decision Trees with a Continuous Label. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.21,No.11, 2009.

[5] J. Ross Quinlan. C4.5 : programs for machine learning. Morgan Kaufmann, San Mateo, Calif., 1993.

[6] G. Hulten, L. Spencer and P. Domingos. Mining Timechanging Data Stream. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pp.97–106, 2001.

[7] S. Nishimura, M. Terabe, K. Hashimoto and K. Mihara. Learning Higher Accuracy Decision Trees from Concept Drifting Data Stream. In Proceedings of the Twenty First International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp.179–188, 2008.

[8] J. Gama, P. Medas, and P. Rodrigues. Learning Decision Trees from Dynamic Data Stream. In Proceedings of the 2005 ACM Symposium on Applied computing, pp.573–577, 2005.

[9] Bernhard Pfahringer, Georey Holmes, and Richard Kirkby. Handling Numeric Attributes in Hoeffding Trees. Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining, pp.296–307, 2008.

[10] T. Minegishi, M. Ise, A. Niimi and O. Konishi. Extension of Decision Tree Algorithm for Stream Data Mining Using Real Data. IEEE, 5th International Workshop on COMPUTATIONAL INTELLIGENCE and APPLICATIONS 2009.

[11] T. Minegishi, M. Ise, A. Niimi and O. Konishi. Comparison with two attribute selection methods using actual data, stepwise procedure in logistic regression analysis and selection by decision tree. Japan Society for Fuzzy Theory and Intelligent Informatics, The 25th Fuzzy System Symposium, 1A2-02 (6 pages in CD-ROM), 2009.

[12] S. Nishimura, M. Terabe, and K. Hashimoto. Decision Tree Induction from Numeric Data Stream. FIT2009, 2009.

[13] P. Domingos, and G. Hulten. VFML - a toolkitfor mining high-speed time-changing data streams. http://www.cs.washington.edu/dm/vfml/, (Access Date: 18 January, 2011).

[14] A. Frank, and A. Asuncion. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/, Irvine, CA: University of California, School of Information and Computer Science, (Access Date: 18 January, 2011).

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and H. Ian. The WEKA Data Mining Software. SIGKDD Explorations, Volume 11, Issue 1, 2009.

