International Journal for Information Security Research (IJISR), Volume 3, Issue 1, March 2013

Proposal of Credit Card Fraudulent Use Detection by Online-type Decision Tree Construction and Verification of Generality

Tatsuya Minegishi[1], Ayahiko Niimi[2]
Graduate School of Systems Information Science, Future University Hakodate[1]
Faculty of Systems Information Science, Future University Hakodate[2]

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 numerous and complex ways. In particular, for data that is generated intermittently, 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 type of decision tree learning called the Very Fast Decision Tree (VFDT) learner, which regards real data as a data stream. We analyze credit card transaction data as a data stream and detect fraudulent use. In recent years, credit card users have increased. However, this also consequently increases the damage caused by fraudulent use. Therefore, the detection of fraudulent use by data stream mining is required. However, some data, such as credit card transaction data, is extremely different from the rate of classes. Therefore, we propose and implement new statistical criteria to be used in a node construction algorithm that implements the VFDT. We also evaluate whether this method can be applied to imbalanced distribution data streams.

1. Introduction

Recent developments in information processing techniques have enabled us to accumulate large-scale data. The need for discovering and utilizing 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 attracted considerable attention. However, with the spread of the Internet and the development of sensor techniques, the complexity of this data is constantly changing, and the increasing amounts of 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 considerable attention. Many classification learning methods have been proposed among which the decision tree learning method is commonly used, because it is fast and the derived description of classifiers is easily interpreted. 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. Credit card transaction data is considered as the data stream. Therefore, it is possible to detect fraudulent use by classifying transaction data using the VFDT. However, among the various data types, there are some data, such as the credit card transaction data discussed in this study, whose characteristics are extremely different. When such data is used in a data stream, some problems can reduce the accuracy of the VFDT [2, 3].

In this study, we propose a node construction algorithm that is applicable to imbalanced distribution data streams. We also implement and evaluate 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 effectiveness of the proposed method by experiments. In Section 5, 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 effective and attractive classification tools, mainly because they produce easily interpretable and well-organized results and are generally computationally efficient and capable of dealing with noisy data. Decision tree techniques build the classification or prediction models on the basis of recursive partitioning of data, which begins with the entire body of data, then splits the data in two or more subsets on the basis of the values of one or more attributes, and then repeatedly splits each subset in finer subsets until the stopping criteria are met [4].

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

In this study, we do not refer to concept drift rather, we discuss the 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 randomly access them. 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 numerous examples accumulate is called an online-type decision tree. A representative example is the VFDT learner.

The VFDT does not accumulate the examples in the 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 in order to reduce 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 meet the statistical criteria.

The statistical criterion called the Hoeffding bounds [9] is used by the VFDT. The examples accumulated in leaf nodes are only a part of all the available examples. Therefore, they can possibly 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 on the basis of 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 bounds 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:

$$\varepsilon=\sqrt{\frac{R^2\ln\left(\frac{1}{\delta}\right)}{2n}} \tag{1}$$

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 bounds, 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 functions, 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, a 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 changes to the construction algorithm of the VFDT. To verify the VFDT change by data sampling, we constructed 10 VFDTs using 10 data sets when constructing the VFDT using credit card transaction data. Because the best-grown decision tree achieves 10% in constructing decision trees by three types of fraudulent use rate experiments in the construction of offline-type decision trees in the existing research [11], we here set the fraudulent use rate to 10%. In offline-type decision trees that are constructed by C4.5, the fraudulent use rate setup is as follows:

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

(a) becomes a decision tree that is 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, however both the fraudulent use rates 0.02% and 0.5% are considerably low to actually classify almost the entire fraud data. Therefore, we used a data set of 10% fraudulent use rate to construct the VFDT.

To evaluate 10-fold cross-validation in all VFDTs and to 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 their size 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 bounds, which is the node construction criterion described in Section 2.1, assumes the data distribution of the data stream to use Gaussian distribution [12]. However, the credit card transaction data described in Section 4.1 contains classes of examples including data streams that do not follow Gaussian distribution. In this case, the accuracy of the constructed VFDT is high, however the actual classification accuracy of the smaller class is almost neglected.

Therefore, we propose the construction of a VFDT that can be applied to imbalanced distribution data streams to improve the calculation of the Hoeffding bounds. We weight the entropy of $G(X_a)$ and $G(X_b)$ using the calculation of information gain

$$\overline{\Delta G()} = \overline{G(X_a)} - \overline{G(X_b)} > \varepsilon \qquad (2)$$

by judging when it grows new branches from leaf nodes. In this study, we define the examples in two classes.

The calculation of entropy using ID3 and C4.5 defines that $freq(C_i,S)$ to the set of examples *S* is the number of examples that is 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:

$$\inf o(S) = -\sum_{i=1}^{2} \frac{freq(C_i,S)}{|S|} \log_2\left(\frac{freq(C_i,S)}{|S|}\right) \qquad (3)$$

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

$$\inf o(S) = -\omega \times \frac{freq(C_1,S)}{|S|} \log_2\left(\frac{freq(C_1,S)}{|S|}\right) - (1-\omega) \times \frac{freq(C_2,S)}{|S|} \log_2\left(\frac{freq(C_2,S)}{|S|}\right) \qquad (4)$$

Therefore, equation (2) becomes

$$\overline{\Delta G'()} = \overline{G'(X_a)} - \overline{G'(X_b)} > \varepsilon \qquad (5)$$

By weighting the proposed method, it normalizes the data distribution and the smaller class is reflected in the learning described in Figure 1.

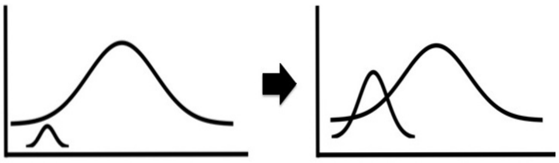

Figure 1. Normalizing data distribution

Therefore, it is classified more correctly than in the proposed method.

In addition, 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. In particular, entropy-based discretization [12] has been adopted. When the VFML discretizes the 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 bounds as $\Delta G'() > \varepsilon$ (equation (4)). Therefore, if it directly compares the Hoeffding bounds ε of the right part, then it compares weighted to non-weighted. Hence, we multiply the information gain of the right part by the average 0.5 of ω and $(1-\omega)$ to two classes, and balance both sides.

4. Experiments

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

4.1. Credit card transaction data

In this study, we consider credit card transaction data, which is real data, as a data stream and perform experiments to detect fraudulent use. Credit card transaction data can be obtained continuously for 24 hours and 365 days. Moreover, the fraudulent use rate is extremely low, namely from 0.02% to 0.05%. Our problem is to detect fraudulent use data in large-scale data; therefore, we can definitely consider credit card transaction data to be imbalanced data stream. In previous research [10], the VFDT classified most fraudulent use as normal use, because the fraudulent use rate was extremely low. In these experiments, we use the data that yielded good performance in C4.5, that is, the offline-type decision tree construction from previous experiments [11].

4.2. Experimental methodology

We compare the results of the VFDT proposed in Section 3 and the conventional method described in Section 2.2 by using the training data in Section 4.1. Moreover, we classify the test data described in Section 4.1 by those VFDTs and compare the results. We compare the classification accuracy, tree size, runtime, and number of fraud rules. We set for the weight of the proposed method in Section 3, because the fraudulent use rate of the re-sampled data is 10%. This weight is the inverse number of data distribution. By comparing the information gain and the Hoeffding bounds in Section 3, the result of is found to be the same as the conventional VFDT.

5. Experimental results

Here, we show the results of each experiment.

5.1. Learning of VFDT

We compare the result of the proposed VFDT in Section 3 and the conventional VFDT in Section 2.2 by using the training data described in Section 4.2. Those results are calculated using 10-fold cross-validation. Table 1 describes the classification accuracy (%), tree size, runtime (s), and number of fraud rules.

Table 1. Accuracy, tree size, runtime, and number of fraud rules.

	Accuracy (%)	Tree size	Runtime (s)	Fraud rules
Proposed method	92.325	106.600	6.722	5
Conventional method	90.851	91.000	5.907	3

The classification accuracy of the proposed method is higher than that of the conventional method. This is because the tree size is increased, thus the classification of the fraudulent use class is improved with the increasing number of fraud rules. As a result, the classification accuracy also improves. In addition, the construction time of the VFDT does not change. In previous research [10], however, we verify the result whose classification accuracy consist of classifying most normal use classes as normal use class. Therefore, we cannot compare whether fraud use class classify correct or not. Thus, we compare them by using the confusion matrix of Tables 2 and 3.

Table 2. Confusion matrix (proposed method)

		Actual classes	
		0 (Normal)	1 (Fraud)
Leaf classes	0 (Normal)	40,174	1,982
	1 (Fraud)	2,145	2,790

Table 3. Confusion matrix (conventional method)

		Actual classes	
		0 (Normal)	1 (Fraud)
Leaf classes	0 (Normal)	40,825	2,174
	1 (Fraud)	1,494	2,598

On comparing the confusion matrices, the number of fraudulent use classes, which are classified as fraudulent use classes described in Table 2, is greater than over the conventional method described in Table 3 by 192. The recall of the fraudulent use class (in this study, we define that the data that is classified as fraudulent use by the VFDT accounts for the total data whose class is fraudulent use) increases 4%.

For comparing the fraud rules of the proposed and conventional methods, Tables 4 and 5 describe the number of fraudulent use classes that is classified as fraudulent use classes and the number of normal use classes that is classified as normal use classes for each method.

Table 4. Number of classes classified by fraud rules (proposed method)

	Fraudulent use class	Normal use class
Fraud rule 1	292	236
Fraud rule 2	98	72
Fraud rule 3	1,343	961
Fraud rule 4	70	58
Fraud rule 5	987	818
Total	2,790	2,145

Table 5. Number of classes classified by fraud rules (conventional method)

	Fraudulent use class	Normal use class
Fraud rule 1	38	15
Fraud rule 2	1,628	1,247
Fraud rule 3	932	232
Total	2,598	1,494

From Tables 4 and 5, for all combinations of the fraud rules of each weight (proposed: 31, conventional: 7), we examine how much data is classified as combinations of fraud rules and how many fraudulent use classes are classified by these fraud rules. The total number of fraudulent use classes that is classified in the fraud rules in the proposed method is higher than that in the conventional method. However, the number of fraudulent use classes that is classified is less than the total number of fraudulent use classes by the conventional method. However, classifying the fraudulent use class is possible by using the number of data that is classified as the more appropriate fraud rules, because the number of fraud rules in the proposed method is higher than that in the conventional method.

In addition, because the weight of the fraudulent use class whose value is w = 0.9 is larger in the proposed method, the number of normal use classes that is misclassified as fraudulent use classes increases. However, it is most important for us to improve the accuracy of the smaller class on the basis of the characteristics of credit card transaction data, as described in Section 4.1. Therefore, we infer that the weighting of the proposed method is effective.

5.2. Test of VFDT

We compare the results of classifying the test data that is described in Section 4.2 using each VFDT that is constructed in Section 5.1. The test data consists of 25,126,264 transactions, 12,280 fraudulent use classes, and 25,113,984 normal use classes.

In these tests, we use the test data whose fraudulent use rate is 0.048%. This rate is a real value in the fraudulent use detection field. We classify the data by using each VFDT that is constructed in Section 5.1.

The conventional method classifies all data as normal use classes. But proposal method can classify 346 data having fraud use class as fraudulent use class correctly. Our problem is to detect an extremely small fraudulent use from a large amount of data in the credit card transaction data field. Therefore, it is difficult to correctly classify a fraudulent use class by using VFDT. However, although the conventional method misclassifies all data whose fraudulent use rate is real in the detecting field, proposal method can classify not a little. Therefore, it can improve the recall of a fraudulent use class. Also, Table 6 describes how much data that the VFDT classifies as a fraudulent use class that is classified according to the five fraud rules described in Table 4.

Table 6. Number of classes classified by fraud rules (proposed method).

	Fraudulent use class	Normal use class
Fraud rule 1	346	145,457
Fraud rule 2	0	0
Fraud rule 3	0	0
Fraud rule 4	0	0
Fraud rule 5	0	0
Total	346	145,457

From Table 6, we confirm that 346 data whose class is fraudulent use and 145,457 data whose class is normal use, which the VFDT misclassifies as a fraudulent use class, are classified by Fraud rule 1. Therefore, we can conclude that Fraud rule 1 classifies a fraudulent use class well.

By weighting when comparing the information gain and the Hoeffding bounds in node construction, we verify that it improves the accuracy in the case wherein a smaller class is not classified well.

Henceforth, we will compare the results using Cumulative Accuracy Profiles (CAP), which is used to evaluate the performance of a prediction model.

5.3. Additional experiments

From the experiments in Sections 5.1 and 5.2, we can verify the effectiveness of our proposed method. Here, we verify the generality of our method by performing similar experiments using data other than credit card transaction data.

The data that we use for the additional experiments is the UCI data set [14], which is widely used as a data archive. We construct each VFDT using the Spambase data set in the UCI and compare the results. The Spambase data set provides information on whether an e-mail is spam mail and contains 1,813 data whose class is spam and 2,788 data whose class is non-spam. We set the weight from 0.1 to 0.9.

As a result, in the case where the weights are 0.1, 0.2, 0.3, and 0.4, the improvement in accuracy is better than the conventional method. From the confusion matrix, although the conventional method correctly classifies 721 data whose class is spam as the spam class, the proposed method correctly classifies 1,620 data.

We can verify the generality of our method, because it is effective when using data other than credit card transaction data.

6. Conclusion and future works

In this study, 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 effectiveness. As a new statistical criterion for a node construction algorithm, we weighted the entropy class by comparing the Hoeffding bounds to the information gain used in splitting the nodes of the conventional algorithm. We believe 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 that is constructed using the UCI data set.

As a result, we can improve the accuracy and the recall of weighted smaller classes with both data sets. In particular, depending on the weight values, the accuracy of the VFDT is the same as that of the offline-type decision tree constructed using a C4.5 algorithm.

However, using credit card transaction data in these experiments, the accuracy of the VFDT and the recall of the weighted class vary, and the results do not conform to 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.

7. Acknowledgment

We thank the employees of INTELLIGENT WAVE INC. for the data, their advice, and the guidance for the experiment

8. 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, pp.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 toolkit for mining high-speed time-changing data streams. http://www.cs.washington.edu/dm/vfml/, (Access Date: January 18, 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: January 18, 2011).

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