

Backward Time Related Association Rule mining with Database Rearrangement in Traffic Volume Prediction

Huiyu Zhou, Shingo Mabu, Kaoru Shimada and Kotaro Hirasawa
Graduate School of Information, Production and Systems
Waseda University
Kitakyushu, Japan

zhy836@toki.waseda.jp; mabu@aoni.waseda.jp; k.shimada@aoni.waseda.jp; hirasawa@waseda.jp

Abstract—In this paper, Backward Time Related Association Rule Mining using Genetic Network Programming (GNP) with Database Rearrangement is introduced in order to find time related sequential association from time related databases effectively and efficiently. GNP is a kind of human brain like evolutionary model which represents solutions as directed graph structures. The concept of database rearrangement to better handle association rule extraction from the databases in the traffic volume prediction problems is proposed. The proposed algorithm and experimental results are also included.

Index Terms—Backwards, Time Related, Data Mining, Genetic Network Programming, Traffic Volume Prediction

I. INTRODUCTION

Nowadays, more and more cars are equipped with automotive navigation systems as GPS becomes popular. The challenge is how to build a real-time system that can handle such massive amount of dynamic and complicated traffic databases and how to predict the future traffic volume so we can give cars the correct navigation guidance to avoid traffic jams and choose the route with the shortest traveling time.

That's why the time related data mining method is needed in these dynamic environments. Data mining is the process of extracting hidden and/or predictive information from large databases. Mining these data efficiently and effectively is too difficult and intricate when using conventional methods, especially for the time related database in dynamic systems.

The obtained relations between data will be represented by association rules as $X \Rightarrow Y$, where X is the antecedent part and Y is the consequent part. A conventional association rule is like the following: "If Section S on the traffic map has high traffic volume, then the Section D also has high traffic volume." But in real-time systems, conventional association rules are not enough to predict the future traffic situation, so, we need the time related association rules like: "If Section S on the traffic map has high traffic volume at time $t = -10$ (10 time units before), then Section D will also has high traffic volume at $t = 0$ (current time)."

In order to extract these sequential association patterns in the time-related databases and represent at what time the event will happen or how long the event will continue, we rearrange the database in this paper to extract important time related

association rules using Genetic Network Programming [1], [2], [3] as a tool to pick up possible candidate rules.

The most popular model in the association rule mining is Apriori algorithm, in which Agrawal et al proposed support-confidence framework [4]. However, Apriori algorithm may suffer from large computational complexity for rule extraction when extracting association rules from dense databases. Many approaches have been proposed to extract association information in various directions, including efficient apriori-like mining methods [5], [6]. The variations of Apriori approach such as the hash-based algorithm have also been studied for efficiency [7], [8].

Genetic Algorithm(GA) [9],[10], proposed by J.H. Holland, has also been applied to data mining research in order to deal with dense databases such as Pittsburgh approach and Michigan approach.

The inherent features of GNP make it possible to work well especially in dynamic environments such as the traffic volume prediction problem. The main task of GNP-based data mining [11], [12] is to allow the GNP individuals to self-evolve and extract association rules as many as possible. But, unlike other evolutionary methods, GNP uses evolved individuals(directed graphs of GNP) just as a tool to extract candidate association rules.

The basic feature of the proposed backward time related association rule mining with database rearrangement in this paper is like the following:

- The database is time-related considering the real-time factor. Thus, the same attribute has different meanings at different time points.
- Extracted rules is backward time related association rules, which means the time delay in the antecedent part are negative numeric tags, which represent the time shift to the current time point $t = 0$, respectively. Backward time related rules are easier for the analysis and pruning of the time related rules.
- In order to adapt to the backward association rule mining, we use database rearrangement method. The database is rearranged to essentially two dimensional data blocks depending on the class of the object attribute. Based on

the object attribute, the blocks are divided to Groups, which increases the efficiency of the whole algorithm.

- The process of searching the database for calculating the support, confidence and chi-squared value is generally one dimensional, but, the searching becomes essentially two dimensional when considering time related databases.
- A simple classification mechanism is combined with the rule extraction system, thus the time related rules can be used to predict the future traffic volume on the traffic networks.
- Another feature of our method is that we can predict not only the traffic jam of a specific section on the road networks, but also can predict the low, middle and high traffic volume of all of the sections on the road networks, which might be useful in future applications.

II. GENETIC NETWORK PROGRAMMING (GNP)

Genetic Network Programming(GNP) is briefly introduced in this section. GNP is an extension of Genetic Algorithm(GA) [9], [10], which represents solutions as directed graph structures [1], [2], [3]. The directed graph of GNP solution aims at imitating human brain functions, i.e., the judgement node imitates IF type conditional judgements, while processing node is corresponding to real actions. The network structure of the judgement node and processing node is in fact building a if-then type conditional judgement and processing model like human brain.

GNP can reuse nodes, thus has compact structures and implicit memory function, which can deal with not only the current information but also the past information implicitly. Also, GNP can find solutions of the problems without bloating compared with Genetic Programming(GP) [13], [14], because of the fixed number of nodes in GNP.

The basic expression of GNP node is shown in Fig.1. This describes the gene of node i , then the set of these genes represents the genotype of GNP individuals. NT_i describes the node type, $NT_i = 0$ means node i is the start node, $NT_i = 1$ represents node i is a judgment node and $NT_i = 2$ represents node i is a processing node. ID_i is an identification number, for example, $NT_i = 1$ and $ID_i = 1$ mean node i is J_1 (Judgment node with ID 1). C_{i1}, C_{i2}, \dots , denote the nodes connected from node i . d_i and d_{ij} are the delay times required to execute the processing of node i and transition from node i to node C_{ij} . The left side of Fig.1 shows the phenotype of the GNP individual, while its corresponding genotype is shown on the right side of Fig.1

As shown in Fig.1, the graph structure of GNP includes three kinds of nodes: start node, judgment nodes and processing nodes. Judgment nodes are the set of J_1, J_2, \dots , which work as if-then type logical decision making functions. On the other hand, processing nodes are the set of P_1, P_2, \dots , which work as a certain kind of action/processsing function. The concrete roles of these nodes are predefined and stored in the library. Once GNP is started up, firstly the execution starts from the

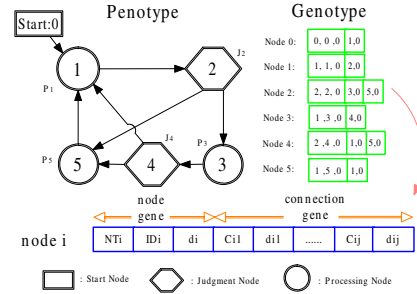


Fig. 1. Basic structure of GNP

start node; consequently, the next node is determined according to the connections and judgment result of the current node.

III. OUTLINE OF THE PROPOSED METHOD

The whole procedure of the proposed algorithm is like the following. The first step is the rule extraction step. In the rule extraction procedure, two kinds of iterations are included, i.e., the outer iteration represents the class association rule mining, which means to extract the class association rules of all the consequent attributes with Attribute Accumulation Algorithm [11], while the inner iteration represents the basic procedure of the time related data mining, which uses GNP individuals to generate the candidate rules, then calculate the support, confidence and chi-squared values of the candidate rules based on the time related databases.

The second step is to use the extracted rules to predict the future events. The selected class is the classification result. If the classification result is the same with the actual situation, then the prediction is correct, otherwise incorrect, and we can obtain the accuracy of the prediction in this way.

A. Structure of Rules

Let $A_i(*) (t = p)$ be an attribute in the database at time p and its value is 1 or 0.

$A_i(*)$ represents $A_i(Low)/A_i(Middle)/A_i(High)$.

Conventional forward time related rules are natural representation of the events happening in sequence like the following:

$$A_1(Low)(t = 0) \wedge A_2(Middle)(t = 2) \Rightarrow$$

$$A_c(High)(t = 5) \text{ or}$$

$$A_3(Low)(t = 0) \wedge A_2(Middle)(t = 4) \wedge A_4(High)(t = 6) \Rightarrow$$

$$A_c(High)(t = 7).$$

Therefore, if we suppose the consequent part is the current time point ($t = 0$) in the above two rules, we can find that $A_2(Middle)(t = 2)$ and $A_2(Middle)(t = 4)$ in fact represent the same event, even though with different time tags.

The situation becomes complicated especially when we consider the pruning and analysis of these kinds of forward rules, because it is difficult to pinpoint the same attribute from thousands of extracted time related association rules. Therefore, we propose backward association rules in this paper.

The proposed method extracts the following backward association rules:

TABLE I
DATABASE BEFORE DISCRETIZATION

| Time | A ₁ | A ₂ |
|------|----------------|----------------|
| 0001 | 8 | 3 |
| 0002 | 4 | 4 |
| 0003 | 8 | 1 |

TABLE II
DATABASE AFTER DISCRETIZATION

| Time | A ₁ | | | A ₂ | | |
|------|----------------|--------|------|----------------|--------|------|
| | Low | Middle | High | Low | Middle | High |
| 0001 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0002 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0003 | 0 | 0 | 1 | 1 | 0 | 0 |

Middle threshold=4; High threshold=7;

$$\text{www } (A_j(*) (t = p) = 1) \wedge \dots \wedge (A_k(*) (t = q) = 1) \Rightarrow (A_c(*) (t = r) = 1)$$
 (briefly, $A_j(*) (t = p) \wedge \dots \wedge A_k(*) (t = q) \Rightarrow A_c(*) (t = r)$)
 Here, $p \leq q \leq r = 0$, the final r always equals 0, and other time points are the relative negative time shifts from the current time, where, $A_c(*)$ indicates the objective attribute. For example: $A_1(\text{Low})(t = -22) \wedge A_2(\text{Low})(t = -6) \Rightarrow A_3(\text{High})(t = 0)$ means that A_1 is Low at 22 time units before and A_2 is Low at 6 time units before, then A_3 is High at the current time unit. These kinds of rules can find the time related relations between attributes and can be easily used in the prediction problems.

B. Database Rearrangement

Traffic prediction is one of the dynamic problems, because the traffic flow in each section of the road networks is changing constantly as time goes on. In order to better predict the traffic flow of the road networks in the future, the database should represent at what time the events in the road section happened or how long the events lasted. Thus, we can have the database shown in Table I.

Here, the time unit represents the conceptual time unit, i.e., it could be one second, one minute or the time for one episode, therefore, the time unit could be flexible enough according to the concrete problem to solve. A_1 and A_2 represent the name of the sections on the traffic networks and data recorded in the database mean that how many cars there are on each section of the map during each time unit.

According to the range of the continuous values of the attributes, we divide the continuous values into three categories, i.e., Low, Middle and High. Supposing that the Middle threshold is 4 and High threshold is 7, the attribute $A = 9$ is ranked as $A - \text{High}$, i.e., $(A - \text{Low}, A - \text{Middle}, A - \text{High}) = (0, 0, 1)$ as shown in Table II.

In order to adapt to the backward time related association rule extraction and increase the whole efficiency of the algorithm, we introduce database rearrangement method in this paper. Now, we explain the database rearrangement depending on the class of the object attribute A_c using Fig.2. If the class *Middle* of the object attribute A_c of the record 0010 is satisfied, its block is defined to include a number of

consecutive records (Time Span) in the time backward way like Fig.2.

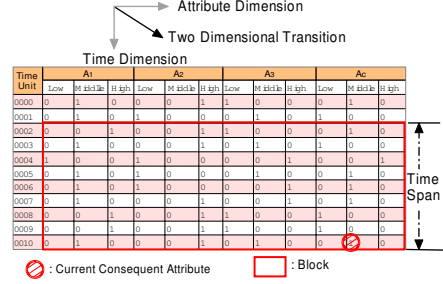


Fig. 2. Database Transitions

The block is described using a rectangular with bold lines in Fig.2, and each block corresponds to a certain class of the objective attributes. The set of the blocks is named Group. As a result, instead of checking the tuples sequentially in the time related database, we rearranged the databases to different groups, Group 0, Group 1 as shown in Fig. 3.

GNP transits using the data in blocks, in other words, each block contains all the data starting backwards from a certain time unit with the length of *MaxTimeSpan*. If the class of the object attribute is satisfied as shown in Fig. 2, its block belongs to Group 1, otherwise, the block belongs to Group 0.

The searching of the backward association rules starts from the bottom line of each block. Since the class of the object attribute is already known for each block in Group 1, we can skip the scanning for consequent part, thus we just have to scan the antecedent part to evaluate each candidate rule.

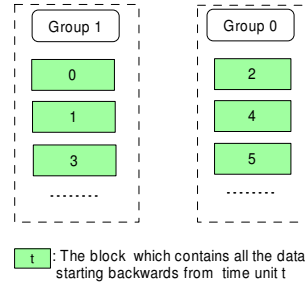


Fig. 3. Database Rearrangement

Generally, the support value of the rule is calculated by $\text{Support} = n/N$, while n represents the number of records in the database which satisfies the rule, and N represents the total number of records in the database. We can see from Fig. 2 that, when calculating the support value for a candidate time related rule with time span of 10, we should consider the Time Span. Therefore, unlike traditional databases, N is not always the total number of records in databases, instead, here N should be calculated by Eq.(1), where, $N_{original}$ is the actual total number of records.

$$N = N_{original} - TimeSpan \quad (1)$$

In order to calculate the support of the rules, we just need to scan only the antecedent part of the database in Group 1 once, since the consequent part is automatically equal to 1 in Group 1, which increases the efficiency largely.

C. Time Related Association Rule Mining with GNP using Backward Transition

The proposed method extracts important time related association rules using GNP. The basic structure of the GNP network structure for class association rule mining with backward transition is shown in Fig.4.

All the items of the consequent part are placed in Consequent Table (CT) shown in Fig.4. This means that GNP individuals are evolved to extract the class association rules for each class of the attributes, i.e., for each traffic volume level of each section of the road networks, and this procedure is repeated for all the consequent items. GNP individual exclusively extracts the association rules about one consequent item in CT, which is called the current consequent item as shown in Fig.4. In addition, a sufficient fixed number of rules for each item in CT are obtained in the proposed method.

P_1 here is a processing node and is a starting point of association rule mining. Each Processing node has an inherent numeric order (P_1, P_2, \dots) and is connected to a judgment node. Yes-side of the judgment node is connected to another judgment node. No-side of the judgment node is connected to the next numbered processing node. The total number of tuples moving to Yes-side at each judgment node is calculated for every processing node, which is fundamental for calculating association rules.

From Fig.4, we can see that each judgment node of GNP is in charge of examining the corresponding attribute. For example, if A_3 satisfies the "Middle" condition, then, moves to the Yes-side of the transition, and "a" is the count of the moves to the Yes side at $A_3(Mid)$. The rules extracted here are backward rules, thus the time delays are all minus one. The time delays are used for the two dimensional search[11], which will be explained later in more detail.

The total number of moving to Yes-side at each judgment node is calculated for every processing node, which is a starting point for calculating criteria for association rules as described above. Fig.5 describes the two dimensional search in more detail, where N is the total number of the search, and a , b and c are the numbers of the search moving to the Yes-side of $A_3(Mid)$, $A_2(Low)$ and $A_1(Mid)$ sequentially. The measurements are calculated by these numbers.

In the proposed method, the searching should consider not only the attribute dimension but also the time dimension, thus the method is two dimensional. That is, not only the attribute but also the corresponding time delay should be considered: For example, as described in Fig.5: the judgment is not merely executed row by row, but the procedure is like the following: first, we pay attention to the final record of the first block in the Group 1 and suppose its time unit is 0005 like Fig.5,

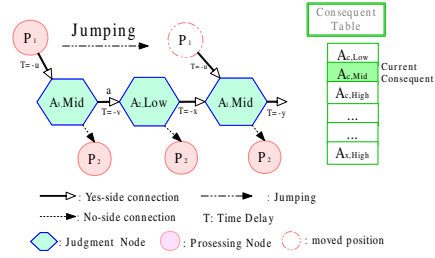


Fig. 4. Basic structure of class association rule mining using GNP

Then, it is checked whether $A_3(Middle) = 1$ at time 0003, because the time delay from P_1 to $A_3(Middle)$ is $T = -2$. If the records satisfy $A_3(Middle) = 1$, then move to check if $A_2(Low) = 1$ at time 0001, since the time delay from $A_3(Middle)$ to $A_2(Low)$ is also $T = -2$. If the value of $A_2(Low)$ at time 0001 is also 1, continue the judgement likewise. If the transition is No-side of the judgement nodes, then the procedure is transferred to the next processing node. After these procedures are carried out for all the processing nodes, another turn of judgements starts from block 2 will begin. This procedure continues until the end of the blocks of the Group 1.

After finishing the checking of Group 1 we can get the support value of each candidate rule without checking the count of consequent part, and when we need to calculate the confidence and chi-squared value, the Group 0 should also be scanned to get the frequency of the antecedent part of the candidate rules in the whole database.

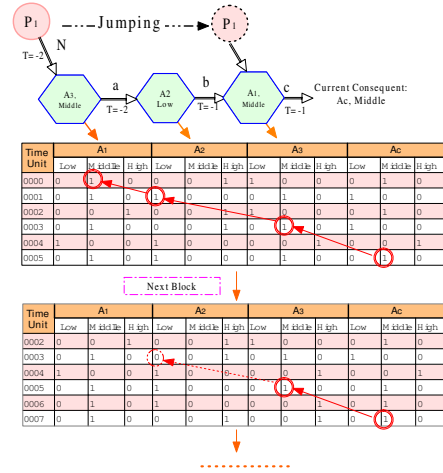


Fig. 5. Two dimensional searching method

D. Important Rules and Genetics Operations

The candidate rules can be considered as important if and only if its support, confidence and chi-squared value is bigger than the minimum support, confidence and chi-squared value,

respectively. χ_{min}^2 , sup_{min} and $conf_{min}$ are initialized for each item of the Consequent Table(CT).

Generations are divided to different rounds, where, Attribute Accumulation Mechanism(AAM)[12] is also applied here to accumulate better attributes round by round in our algorithm, thus we can find better and fitter attributes in GNP in an exploitative way.

Now, we define the concept of multiplicity, which means that the rules containing many different attributes are preferable in order to maintain the diversity of the whole evolution process. Therefore, the fitness function of GNP is defined as:

$$F = \sum_{r \in R} \{\chi^2(r) + 10(n_{ante}(r) - 1) + \alpha_{new}(r) + \alpha_{mult}(r)\}.$$

The symbols are as follows:

R : set of suffixes of important association rules which satisfy the importance requirements in GNP individuals.

$\chi^2(r)$: chi-squared value of rule r .

$n_{ante}(r)$: the number of attributes in the antecedent of rule r .

$\alpha_{new}(r)$: constant defined as

$$\alpha_{new}(r) = \begin{cases} \alpha_{new}, & \text{if the rule } r \text{ is new} \\ 0, & \text{otherwise} \end{cases}$$

$\alpha_{mult}(r)$: constant defined as

$$\alpha_{mult}(r) = \begin{cases} \alpha_{mult}, & \text{if the rule } r \text{ has many} \\ & \text{different attributes} \\ 0, & \text{otherwise} \end{cases}$$

$\chi^2(r)$, $n_{ante}(r)$, $\alpha_{new}(r)$ and $\alpha_{mult}(r)$ are concerned with the importance, complexity, novelty and diversity of rule r , respectively. At each generation, GNP individuals are replaced with new ones by the selection policy and genetic operations. We used four kinds of genetic operators, i.e., uniform crossover, mutation for functions, mutation for connections and time delays of judgement nodes, respectively.

IV. SIMULATION

In this section, the effectiveness and efficiency of the proposed method are studied by a simple traffic simulation. Unlike other methods in the traffic prediction of recent years, we not only aim at predicting the traffic jam on a specific section of the road networks, but also, we are interested in providing the whole traffic prediction for all of the sections on the road networks so that the navigation system can refer to this kind of information for the calculation of the optimal route.

A. Traffic Simulator

Each section between two intersections in the road networks has two directions, and we assume each direction of the section represents different literals, i.e., items or attributes. The traffic simulator used in our simulations consists of the road model with 7×7 roads like Fig.6, i.e., each section has the same length, all cars have the same speed for simplicity, and the shape of the total road is like a grid network. Time shift in road setting in Fig.6 represents the time delay of the traffic lights between neighboring intersections.

Actually the cars on the road networks do not have the same speed. Every time unit, all the cars can move forward by length 1 if and only if there exist spaces before them, thus the actual speed of all cars are influenced by the traffic lights or traffic jams. For example, if a car encounters the red light or traffic jam, it has to wait until the red light period passes or all the hindrances before it are removed. Therefore, the cars have different speeds depending on the concrete traffic situations.

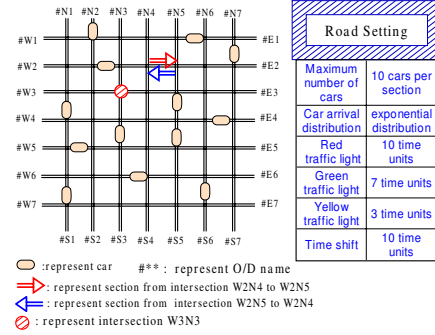


Fig. 6. Road model used in simulations

B. Simulation on Number of Rules

The number of rules stored in the rule pool is compared between the proposed method with backward database rearrangement and conventional basic time related data mining method without backward database rearrangement. Each round has the same number of generations of 50 and the chosen set size for AAM is 100 [11], [12].

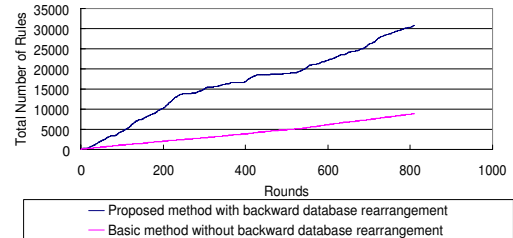


Fig. 7. Total number of rules extracted.

Fig.7 shows the total number of rules obtained in the rule pool versus round number [11], [12]. In the conventional method, databases are not rearranged before rule extraction, while database rearrangement is done in the proposed method. We can see from Fig.7 that the proposed method can extract important class association rules more efficiently, compared with the conventional one.

The reason for the proposed method can get larger number of rules than the conventional method is that the satisfaction of the object class is guaranteed before the checking in Group 1, thus the consequent part do not need to be checked. As a

result, we just need to check all the possible combination of the antecedent part in Group 1 and Group 0 to get the overall frequency of the candidate rules. Database rearrangement is in fact a kind of data preprocessing which increases the whole efficiency of the program.

C. Simulation on Prediction

In order to check the effectiveness of the extracted rules, we tested the accuracy of the classification using the same classifier in [11], [12]. A 5-fold cross validation was carried out for 5 different databases, which are all generated by using the simulator described above, and we obtained the average accuracy shown in Table III, where D represents each database.

The accuracy is defined as the following: if the traffic prediction result of the section at time t is "Low" and the real traffic of this section at time t is exactly "Low", then the accuracy is 100%. The Low/Middle/High accuracy means the accuracy when the real traffic is Low/Middle/High, respectively.

TABLE III
CLASSIFICATION ACCURACY OF CROSS VALIDATION

| D | Overall | Low | Middle | High |
|---|---------|-------|--------|-------|
| 1 | 83.11 | 63.38 | 89.17 | 80.51 |
| 2 | 84.12 | 64.63 | 88.48 | 83.70 |
| 3 | 83.09 | 65.03 | 88.25 | 82.22 |
| 4 | 84.82 | 66.44 | 88.80 | 80.67 |
| 5 | 83.03 | 67.41 | 89.14 | 80.41 |

The prediction accuracy of the Low traffic volume is relatively low because the prediction are done using a relative busy traffic network, where, most of the records shows "Middle" or "High", thus, the information related to the "Low" association rules is missing from the database.

D. Simulation on Time Efficiency

This simulation is devoted to analyze the time efficiency of the extraction in the proposed method. We compared the average time for extraction to get a fixed number of association rules for the class of the object attributes. The conventional one in this simulation uses 50 generations for each round, which is the same as our proposed database rearrangement method. The result is shown in Fig. 8.

We can see from Fig. 8 that even though the database rearrangement and grouping may consume some extra time, the average extraction time in data rearrangement is still better than conventional one, since we just need to examine only the antecedent part for each rule.

V. CONCLUSION

In this paper, backward time related association rule mining in traffic prediction using Genetic Network Programming with database rearrangement has been proposed. It is clarified from simulations that the proposed method can extract important backward time-related association rules for each class of the consequent attributes efficiently. These rules are used to decide to which class the time-related data belong. From simulations results, we have also found that the proposed method can be

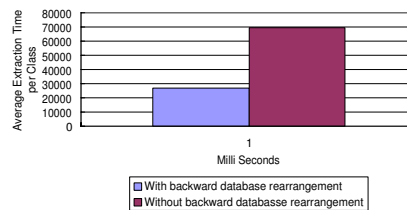


Fig. 8. Average time for extracting rules.

used in traffic prediction problems effectively. We are now studying further improvements of our method in terms of more efficient association rules mining and classification to deal with rare cases and also have a plan to apply our method to more realistic and large databases using real time traffic datasets.

REFERENCES

- [1] S. Mabu, K. Hirasawa and J. Hu, "A Graph-Based Evolutionary Algorithm: Genetic Network Programming(GNP) and Its Extension Using Reinforcement Learning", *Evolutionary Computation*, MIT press, Vol. 15, No.3, pp.369-398, 2007.
- [2] T. Eguchi, K. Hirasawa, J. Hu and N. Ota, "A study of Evolutionary Multiagent Models Based on Symbiosis", *IEEE Trans. on Syst., Man and Cybernetics - Part B -*, Vol.36, No.1, pp.179-193, 2006.
- [3] K. Hirasawa, T. Eguchi, J. Zhou, L. Yu and S. Markon, A Double-Deck Elevator Group Supervisory Control System Using Genetic Network Programming, *IEEE Trans. on Systems, Man and Cybernetics, Part C*, Vol. 38, No. 4, pp. 535-550, 2008/7.
- [4] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", In *Proc. of the 20th VLDB Conf.*, pp.487-499, 1994.
- [5] T. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen and A.I. Verkamo, "Finding Interesting Rules from Large Sets of Discovered Association Rules", In *Proc. of the Third Int'l Conf. on Information and Knowledge Management*, pp.401-408, Nov.1994.
- [6] A. Savasere, E. Omiecinski and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases", In *Proc. of the 1995 Int'l Conf. on Very Large Data Bases*, pp. 432-443, Sept.1995.
- [7] J. S. Park, M. S. Chen and P. S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules", In *Proc. of the 1995 ACM SIGMOD Conf.*, pp.175-186, 1995.
- [8] X. Wu, C. Zhang and S. Zhang, "Efficient Mining of Both Positive and Negative Association Rule", *ACM Transactions on Information Systems*, Vol.22, No.3, pp.381-405, 2004.
- [9] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.
- [10] D. E. Goldberg, *Genetic Algorithm in search, optimization and machine learning*, Addison-Wesley, 1989.
- [11] H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, Time Related Association Rules Mining with Attributes Accumulation Mechanism and its Application to Traffic Prediction, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 12, No. 5, pp. 467-478, 2008/7.
- [12] H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, "Time Related Association Rules Mining with Attributes Accumulation Mechanism and its Application to Traffic Prediction", In *Proc. of the IEEE Congress on Evolutionary Computation*, pp. 305-311, 2008/6.
- [13] J. R. Koza, *Genetic Programming, on the programming of computers by means of natural selection*, Cambridge, Mass., MIT Press, 1992.
- [14] J. R. Koza, *Genetic Programming II, Automatic Discovery of Reusable Programs*, Cambridge, Mass.: MIT Press, 1994.