Speed-up Technique for Association Rule Mining Based on an Artificial Life Algorithm

Masaaki Kanakubo
Shizuoka Institute of Science and Technology
2200-2 Toyosawa, Fukuroi-city,
Shizuoka Pref. 437-8555, Japan
kanakubo@cs.sist.ac.jp

Masafumi Hagiwara
Keio University
3-14-1 Hiyoshi, Kohoku-ku,
Yokohama 223-8522, Japan
hagiwara@soft.ics.keio.ac.jp

Abstract

Association rule mining is one of the most important issues in data mining. Apriori computation schemes greatly reduce the computation time by pruning the candidate itemset. However, a large computation time is required when the treated data are dense and the amount of data is large. With apriori methods, the problem of becoming incomputable cannot be avoided when the total number of items is large. On the other hand, bottom-up approaches such as artificial life approaches are the opposite to of the top-down approaches of searches covering all transactions, and may provide new methods of breaking away from the completeness of searches in conventional algorithms. Here, an artificial life data mining technique is proposed in which one transaction is considered as one individual, and association rules are accumulated by the interaction of randomly selected individuals. The proposed algorithm is compaired to other methods in application to a large-scale actual dataset, and it is verified that its performance is greatly superior to that of the method using transaction data virtually divided and that of apriori method by sampling approach, thus demonstrating its usefulness.

1. Introduction

Association rule mining is one of the most important issues in data mining, and there is a very high demand in industry as extraction of good association rules is directly related to sales[1, 2, 3]. However, association rule mining requires consideration of combinations of elements (items) comprising the rule, the number of which increases exponentially and has enormous computational cost. Apriori computation schemes greatly reduce the computation time by pruning the candidate itemset[4]. However, a large computation time is required when the treated data are dense (i.e., there is a large item count) and the amount of data is large[5]. Therefore, various speed-up techniques have been studied.

In terms of implementation, studies have been conducted on parallel methods, which divide transaction data onto different processors[6, 7]. In terms of algorithms, the focus of study is how to efficiently calculate support of itemsets with element count 2, which has the highest computational cost, using a simple apriori scheme. Various methods have been reported, such as methods to simplify calculation of support as in the FP-tree algorithm by expressing the itemset with at least minimum support using tree-construction[8, 9, 10] and methods using transaction data virtually divided beforehand to enable pruning of itemsets of element count 2 from support data of items of element count 1[11, 12, 13].

However, although the former can reduce the time taken to count transaction data of each itemset, the amount of computation involved in tree construction for all candidate itemsets is huge, and the latter has a poor effect unless multiple-partition data are divided so that there is a bias in the items contained in the partitioned data. With apriori methods, the problem of becoming incomputable cannot be avoided when the total number of items is large.

On the other hand, artificial life approaches, which have made a number of advancements in recent years, aim to obtain higher-order (multiple optimal or semi-optimal) solutions by utilizing the phenomenon of emergence in which complex phenomena are formed through mutual interaction of artificial life individuals based on simple rules[14, 15]. Bottom-up approaches such as them are the opposite to of the top-down approaches of searches covering all transactions, and may provide new methods of breaking away from the completeness of searches in conventional algorithms.

As a process analogous to artificial life, the bottom-up association rule mining method using Genetic Algorithms

0-7695-3032-X/07 $25.00 © 2007 IEEE
DOI 10.1109/GrC.2007.103

has been proposed[16]. These methods aim to achieve an efficient search by examining the neighborhood rules with changes, obtained by chance, in association rules with high support (mutation) and combination (crossover). For association rule mining, however, support may change greatly in many cases with only minor rule changes, and other solutions in the neighborhood of a good solution (i.e., with similar rules) may not necessarily be good solutions themselves. In general, neither Genetic Algorithms nor the Tabu search method is effective for a search space with such a complex configuration.

Here, an artificial life data mining technique is proposed in which one transaction is considered as one individual, and association rules are accumulated by the interaction of randomly selected individuals. Specifically, this technique is characterized by how the candidate association rules are generated by focusing on a pair of individuals with more than a certain number of common itemsets within the population of N randomly selected individuals. By avoiding the vast computational costs associated with exhaustive examination of all itemsets and by generating candidate rules using itemsets that appeared in the 2 individuals, it is possible to examine only the candidates with higher accuracy compared to methods using randomly generated rules.

Below,section 2 describes the overview of conventional and proposed methods.Section 3 presents a computer simulation in order to investigate the effectiveness of the proposed algorithm, and discusses the results.

2. Overview of conventional and proposed methods

2.1. Apriori algorithm by sampling

In the apriori algorithm, all the transactions of all the itemsets generated (including those generated after pruning) in the intermediary steps are examined to obtain support. Thus, the computation time is proportional to the number of transactions. In the apriori by sampling approach, a certain number of transactions are sampled using a random number, and the apriori method is applied to that set.

2.2. Method using virtual partition data

When all transactions are divided into two without duplication, suppose the number of appearance of all itemsets of item count 1 was as shown in Table 1. If the number of appearances that fulfills minimum support is 10, it is clear from the data in this table alone, and without referring to the transaction data, that the set (B, C) is minsup or less. The method that uses virtual partition data omits enumeration of the most time-consuming set with item count 2 by apriori using such a method. It is possible to perform similar calculations even when the item count increases, but when partitioned data does not have a bias, as shown in Table 1, the effect is small.

Table 1. Example of omission by virtual partition data

Itemset	A	B	C	D
NA in partition data 1	13	12	1	2
NA in partition data 2	3	3	12	14
NA in total	16	15	13	16

NA = Number of appearances

2.3. Proposed Method

In algorithm proposed in this paper, one transaction datum is considered as one individual. A generation is formed by a certain number of individuals. Each individual records the combination generated from the common items in an "Individual Logbook" only when there is more than a constant number of common items with other individuals. In addition, there is a "Common Logbook" that summarizes the record of the population. In addition, there is a "Past Logbook" that summarizes the records through the generations. Once a certain number of generations has passed, it is considered that a "Civilization" has come to an end, and a new civilization is started. The record of the Past Logbook from each civilization is summarized in the "Final Logbook".

A flow chart of the proposed method is shown in Figure 1.

(1) *N* transactions are selected at random from all transactions to form the initial population.

(2) For all combinations of 2 individuals in the population, extract combinations that have more than a certain number of common items.

(3) From each item combination obtained in (2), all the subsets with 2 or more and less than a certain number of element counts are extracted as candidate itemset, and are recorded in each individual's Individual Logbook as "support count 2". Here, when the same itemset has already been recorded in the logbook of a certain individual, the appropriate support count is increased by 1.

(4) The information exchange in processes (2) and (3) is completed for all combinations of 2 individuals in the population, and the itemset and support count of each

individual's Individual Logbook are recorded sequentially in the generation's Common Logbook. Here, when the same set has already been written by other individuals, another individual is not recorded.

(5) Itemsets and support counts recorded in a particular generation's Common Logbook are copied to the Past Logbook, which is recorded for many generations. Here, only the support counts obtained in that generation are added when there are sets left by the ancestors.

(6) *N* transactions are selected randomly from all transactions to make up the next generation, and returns to (2).

(7) Once processes (2) to (6) are repeated, and a certain number of generations has passed and a civilization has come to an end, the itemsets with support more than the minimum support are recorded in the Final Logbook. Here, the calculation is done as support for each set=support count/number of individuals $\times$ number of generations. When the same set is already present in the Final Logbook, it is not recorded. This is followed by a return to the process in (1).

(8) Once processes between (1) and (7) have been repeated a certain number of times, all association rules generated from frequent itemsets in the Final Logbook are checked to determine whether they satisfy the minimum confidence, to obtain association rules that satisfy the condition.

The logbook is divided to restrain the computational costs of merging and sorting a large-scale logbook, and to avoid recording itemsets with low support counts. In the proposed method, when common itemsets of the 2 individuals are being obtained, it is possible to focus on extraction of the long itemsets from the start, by adding restrictions to limit extraction of those with more than a certain number of element counts. In this way, evolution doesn't happen between "generations", but between "civilizations" and there is no crossover or mutation.

3. Evaluation of the Experiment

3.1. Items compared to the Experimental data

The transaction data used for evaluation of the experiment were raw purchase data collected between 1999 and 2000 in a retail store in Belgium, and are often used for evaluation in data mining experiments[17]. This is a large-scale actual dataset consisting of 21,338 transactions, a total number of items of 10,357, and the item count of the longest transaction is 75. These data were used for comparison using minimum support set to 0.0027 and 0.0100, and

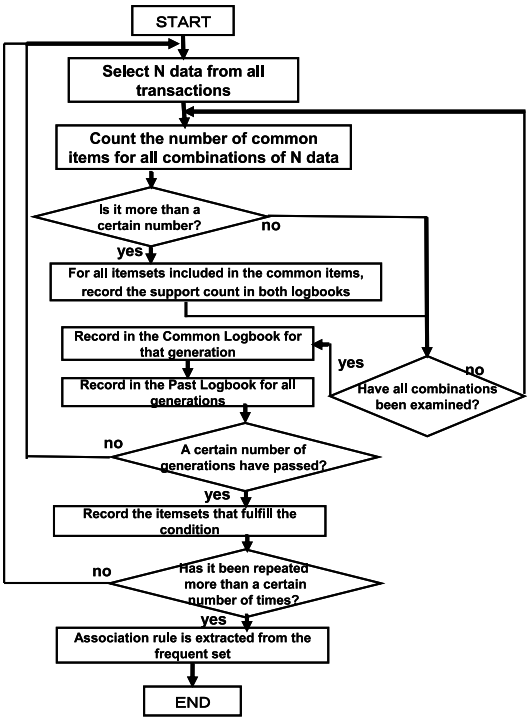

Figure 1. Flow of the process in the proposed method

minimum confidence set to 0.1 for both. The experiment first extracted all association rules that satisfied the condition using apriori, and then the various methods, including the proposed method, were compared with regard to computation time, as well as how many association rules obtained using apriori can be obtained using other methods, and the differences between apriori with regard to support and confidence. Comparisons were also performed according to the length of antecedent and consequent of the association rule combined. (An association rule is of the form $X \Rightarrow Y$, the confidence of the rule is the conditional probability of Y given X, Pr($Y \mid X$), and the support of the rule is the prior probability of X and Y, Pr(X and Y).)

The degrees of coincidence of support and confidence were obtained using the following equation:

$$fitness = \sum_{i=1}^{n} 2.0 - (|\Delta sup_i| + |\Delta conf_i|) \qquad (1)$$

Here, *n* is the total number of rules obtained by a particular method. sup_i is the difference between support of the *i*th rule extracted using that method, and the support of the same rule extracted by apriori. $conf_i$ is the difference between confidence of the *i*th rule extracted using that method, and the confidence of the same rule extracted by

apriori. The values of both support and confidence were between 0 and 1, and thus the sum of the absolute value of the 2 differences should be below 2. Thus, we subtracted 2 from the sum of the absolute value of the differences, and the degree of coincidence will become larger with smaller differences. When the same rule does not exist, the point is not added.

3.2. Experimental parameters in other methods

Three other methods were performed for comparison: apriori for sampled transaction data, the method using virtual partition data described previously (13), and the method that virtually partitions the sampling data. For sampling, 4,000 data were selected randomly from all transactions in each method.

For virtual partition, all transactions were divided into 50 partial transactions. To determine to which subset a transaction should belong, the remainder of the entire item numbers included divided by 50 are counted, and the transaction will belong to the subset with the largest remainder. Therefore, the probability of different items belonging to different subsets will be higher, and the pruning effect from virtual partitioning can be expected. Pruning by virtual partition was performed for all combinations composed of 2 items above the minimum support. In addition, in examination of the combination with item count 3 or more, pruning by apriori was performed after pruning by virtual partition.

3.3. Experimental parameters in the proposed method

Experimental parameters for the proposed method are shown in Table 2. The capacity of the logbook indicates how many rules can be recorded in the logbook, and once the capacity is full, the recorded data are overwritten. The minimum common item indicates how many common items there should be between 2 individuals exchanging information to perform rule extraction. These parameters were determined in advance through trial and error.

3.4. Comparison of rule extraction results

The number of rules of length extracted and, for methods not using apriori, the degree of coincidence with the rule of apriori, are shown in Table 3 for the 2 types of minimum support of each method, including the proposed method, that satisfies the conditions. The length of the rule here indicates the item count of the antecedent and consequent combined. The results of the method that requires the proposed method and random sampling are both the mean values of 10 trials. Here, AP indicates apriori, S-AP indicates apriori to the sample, PART indicates a virtual partition method, S-PART indicates virtual partition method to the sample, and PROP indicates the proposed method. The numbers below each abbreviation indicate the number of rules extracted, and the fraction on the right shows the corresponding degree of coincidence.

Table 2. Experimental parameters of the proposed method

Number of individuals forming one generation	50
Number of repetitions for the generation	20
Number of repetitions for the civilization	1000
Capacity of the Individual Logbook	300
Capacity of the Common Logbook	1000
Capacity of the Past Logbook	3000
Capacity of the Final Logbook	2000
Minimum number of common items	5

Total number of rules extracted (including those not satisfying the condition) and computation time (s) of each method are shown in Table 4.

Each of the methods other than apriori succeeded in extracting roughly the same rule as apriori. The proposed method only extracted about half of the short association rules of which the combined lengths of the antecedent and consequent is 2, but the majority of rules with combined length of more than 3 were extracted. With minimum support of 0.0027 for the rule with the longest length of 5, more rules were extracted by the proposed method as compared to the methods using sampling, and all the association rules were extracted successfully. We believe that this was because the proposed method extracts the rules when 5 or more items are common when making combinations of 2 individuals, which facilitates the extraction of long association rules. The differences in support and confidence of the proposed method as compared to other methods were not significant.

When comparing the execution times of each method, the difference in number of extraction rules was about 10-fold for minimum support of 0.0100 and 0.0027, and a large amount of time was taken especially for apriori. For example, with minimum support of 0.0027, the number of com-

Table 3. Number of rules and degree of coincidence extracted by rule count and each of the methods

Length of rule	AP rule count	S-AP	DC	PART	DC	S-PART	DC	PROP	DC
Minimum support = 0.0100									
2	90	86	169.5	90	180	81	159.4	48	67.9
3	144	129	254.8	144	288	138	271.9	129	187.2
4	79	70	138	79	158	77	152.4	79	112.5
TOTAL	313	285	562.3	313	626	296	583.6	256	367.6
Minimum support = 0.0027									
2	784	659	1280.6	784	1568	667	1295	373	557.4
3	1281	1043	2030	1281	2562	1015	1971.4	956	1449.8
4	732	575	1121.7	732	1464	557	1084.7	718	1096.1
5	116	110	214.5	116	232	90	173.3	116	181
TOTAL	2913	2387	4646.5	2913	5826	2329	4524.4	2163	3284.3

DC=Degree of coincidence

Table 4. Computation time for each method and total number of rules extracted

Method	CT	NR	CT	NR
	Minsup 0.0100		Minsup 0.0027	
AP	242.8	313	1480.7	2913
S-AP	156.6	327	984.4	3324
PART	155.7	313	773.2	2913
S-PART	163.7	372	285.9	3300
PROP	13.1	704	33	15994

CT=Computation time
NR=Number of rules
Minsup=Minimum support

binations with an item count of 2 that require apriori to read all transactions to determine whether it is above the minimum support was 198.765. In addition, the number of combinations with an item count of 4 examined to determine whether pruning can be performed was 22,533,126. On the other hand, for methods using virtual partition data, the time required for minimum support was about half that of apriori in either case. This was due to the decrease in number of combinations with item counts of 2 from 198,765 to 60,851 through virtually partitioning the data. The partition method grouped using the remainder of the item number was effective. The method by sampling also succeeded in greatly reducing the computation time.

Even compared with these methods, the proposed method obtained results in a remarkably short time, which was approximately 1/10 to 1/20 those of the other methods. Even when the artificial life approach is used, it takes a long time for trial and error to find a common rule when using a method that examines rules at random between 2 individuals to determine whether they are common. In the proposed method, 2 individuals with more than a certain length of common items are searched from the start, and the number of times the combination generated from these common items appears is counted. This recording method has less trial and error, which is believed to be responsible for the observed time reduction. Even for data mining of this size, with the maximum length of antecedent and consequent combined of 5 items, and the number of rules to be extracted is 2,913, apriori requires a long time. As the number of combinations increases exponentially as the item count increases, we believe that association rule can only be extracted by a bottom-up method, as in the proposed method, if the data are of larger scale.

One problem for the proposed method is that many noise rules that actually do not satisfy minimum support and minimum confidence are extracted. This is because when a frequent itemset is obtained in the proposed method, the calculation is conducted assuming that the number of individuals $\times$ number of generations (1,000 in this experiment) is the number of all transactions, which is an extremely small amount. However, a great deal of computation time will not be necessary to obtain the correct value from these noise rules with use of virtual partition tables, and therefore this should not be considered a major flaw.

4. Conclusions

In this paper, an artificial life data mining technique is proposed in which one transaction is considered as one individual, and association rules are accumulated by the interaction of randomly selected individuals. Specifically, this technique is characterized by how the candidate association

rules are generated by focusing on a pair of individuals with more than a certain number of common itemsets within the population of *N* randomly selected individuals. By avoiding the vast computational costs associated with exhaustive examination of all itemsets and by generating candidate rules using itemsets that appeared in the 2 individuals, it is possible to examine only the candidates with higher accuracy compared to methods using randomly generated rules.

The proposed algorithm is compaired to other methods in application to a large-scale actual dataset, and it is verified that its performance is greatly superior to that of the method using transaction data virtually divided and that of apriori method by sampling approach. Even compared with these methods, the proposed method obtained results in a remarkably short time, which was approximately 1/10 to 1/20 those of the other methods. As the number of combinations increases exponentially as the item count increases, we believe that association rule can only be extracted by a bottom-up method, as in the proposed method, if the data are of larger scale.

References

[1] T.Terano, "Perspectives on KDD and Data Mining Tools", Journal of The Japanese Society for Artificial Intelligence, Vol.12, No.4, pp.521-527, 1997 (in Japanese).

[2] H.Kawano, "An Overview of Knowledge Discovery in Databases", Journal of The Japanese Society for Artificial Intelligence, Vol.12, No.4, pp.497-504, 1997 (in Japanese).

[3] M.Kitsuregawa, "Mining Algorithms for Association Rules", Journal of The Japanese Society for Artificial Intelligence, Vol.12, No.4, pp.513-520, 1997 (in Japanese).

[4] Agrawal R., Imielinski T., Swami A.:"Mining association rules between sets of items in large databases, " Proc. of the ACM SIGMOD Conference on Manegement of Data, Washington D.C., 1993.

[5] T.Uno and H.Arimura, "An Efficient Method for Finding Association Rules from Large Scale Database Based on Closed Pattern Enuneration", Proceedings of the Institute of Statistical Mathematics, Vol.53, No.2, pp.317-329, 2005 (in Japanese).

[6] Agrawal R., Shafer J.C., "Paralell Mining of Association Rules" IEEE Transactions on Knowledge and Data Engineering, Vol.8, No.6, December 1996. Expanded version available as IBM Research Report RJ 10004,January 1996.

[7] T.Shintani, M.Kitsuregawa, "Hash Based Parallel Algorithms for Mining Association Rules" Proceeding of IEEE Fourth International Conference on Parallel and Distributed Information Systems, pp.19-30, 1996.

[8] Han J., Pei J., Yin Y., "Mining frequent patterns without candidate generation", Proc. SIGMOD, pp.1-12, ACM, 2000.

[9] Coenen F., Goulbourne G., Leng P.H., "Computing association rules using partial totals", Proc. PKDD, pp.54-66, Springer, 2001.

[10] E.Iwahashi, H.Yamana, "Parallel FP-growth Algorithm for Frequent Pattern Mining", Technical Report of IEICE, DE2003-50, pp.109-114, 2003 (in Japanese).

[11] Cheung D.W., Ng V.T., Fu A.W., Fu Y., "Efficient Mining of Association Rules in Distributed Databases" Special Issue in Data Mining, IEEE Transaction on Knowledge and Data Engineering, IEEE Computer Society, Vol.8, No.6, pp.911-922, 1996.

[12] Manning A.M., Keane J.A., "Inducing load balancing and efficient data distribution prior to association rule discovery in a parallel environment" Euro-Par 99:Proc. 5th Int'l European Conference, pp1460-1463,Toulouse, France, August/September, 1999.

[13] T.Shinoda, H.Matsuo, "High-Speed Association Rule Mining using Hash with Minimum Support", IPSJ SIG Notes, Vol.2002, No.22, pp.7-12, 2002 (in Japanese).

[14] Langton C.G, "ARTIFICIAL LIFE ", Addison-Wesley, 1991.

[15] B.Kimoto, M.Hagiwara, "Extraction of Dynamically Changing Relations of Concepts and the Long-term Prediction by an Artificial Life Approach", Journal of Japan Society for Fuzzy Theory and Systems, Vol.11, No.6, pp.145-152, 1999 (in Japanese).

[16] K.Shimada, K.Hirasawa, T.Furuzuki, "Association Rule Mining Using Genetic Network Programming", Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, Vol.18, No.6, pp.881-891, 2006 (in Japanese).

[17] http://www.cs.rpi.edu/ zaki/FIMI/data/