Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 867149, 9 pages
http://dx.doi.org/10.1155/2014/867149

Research Article

A Genetic Algorithm Based Multilevel Association Rules Mining for Big Datasets

Yang Xu,[1] Mingming Zeng,[1] Quanhui Liu,[1] and Xiaofeng Wang[2]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, China*
[2] *Institute of Computing Technology, Chinese Academy of Sciences, China*

Correspondence should be addressed to Yang Xu; xuyang@uestc.edu.cn

Received 1 July 2014; Revised 5 August 2014; Accepted 14 August 2014; Published 26 August 2014

Academic Editor: Shifei Ding

Multilevel association rules mining is an important domain to discover interesting relations between data elements with multiple levels abstractions. Most of the existing algorithms toward this issue are based on exhausting search methods such as Apriori, and FP-growth. However, when they are applied in the big data applications, those methods will suffer for extreme computational cost in searching association rules. To expedite multilevel association rules searching and avoid the excessive computation, in this paper, we proposed a novel genetic-based method with three key innovations. First, we use the category tree to describe the multilevel application data sets as the domain knowledge. Then, we put forward a special tree encoding schema based on the category tree to build the heuristic multilevel association mining algorithm. As the last part of our design, we proposed the genetic algorithm based on the tree encoding schema that will greatly reduce the association rule search space. The method is especially useful in mining multilevel association rules in big data related applications. We test the proposed method with some big datasets, and the experimental results demonstrate the effectiveness and efficiency of the proposed method in processing big data. Moreover, our results also manifest that the algorithm is fast convergent with a limited termination threshold.

1. Introduction

Exploring knowledge in *Big data* is appealing in the state of the art of data mining research [1]. Due to its high volume and complexity, resourceful domain knowledge or hidden patterns are potentially useful for human decision support [2]. It is especially for the case of multilevel association rule mining approaches to discover interesting relations among data elements with multiple levels of abstractions. Successful applications include spatial data analysis [3], emergency event analysis [4], sensor network data mining [5], and gene ontology mining. However, most existing multilevel association rules mining algorithms rely on exhaustive scans of the database to find frequent patterns across different abstraction levels, such as the most renowned Apriori algorithm [6] and Frequent Pattern tree algorithm (FP-tree) [7]. When the dataset scales up, those algorithms will suffer for the excessive computation cost and the system will retard due to its heavy scan of the large database. When the algorithms are used in big data applications, the bottleneck becomes more prominent. For example, in gene ontology domain, the annotations have rapidly grown to more than 80 million by 2012. If considering the complicated relationships between gene items in various hierarchical levels, the complexity of mining association rules in multilevels has been classified as NEXP-COMPLETE [8]. Therefore a fast multilevel association rules mining algorithm for big datasets that is scalable and can be performed in parallel computation environment becomes imperative.

In this paper, we made our initial effort toward this issue by building a genetic algorithm (GA) based heuristic method for effective multilevel association rules mining in big datasets. By taking advantage of the genetic algorithm, which can efficiently find multiple solutions concurrently in a large multidimensional problem without performing exhaustive searches, our proposed method can improve the mining performance while keeping a desired accuracy but avoiding the exhausting enumeration on association rule candidates. In summary, there are three major contributions.

First, to make our GA-based approach possible, we design a new tree like encoding schema to model the genetic candidates of multilevel association rules so that a feasible implementation of genetic operators can be built. This representation model is based on the application's domain knowledge, where each attributes (items) to be mined can be briefly illustrated as a catalog tree. Then, each valid multilevel association rule can be modeled as a subtree of the catalog tree, but each leaf node should be assigned with a binary number to indicate whether it is an antecedent or a consequent in the rule.

Next, based on the encoding schema, we build its unique GA-based operators to make the multilevel association rules mining possible: (1) individual initiation function to build the subtree with valid multilevel association rule representations; (2) crossover function that allows subtree to cross over at one of their common nodes to produce new generations; (3) selection function which is based on our designed fitness function to select stronger association rules.

As the third contribution, based on our analysis of our fitness function design, we have found that our GA-based method is adaptive and robust as its termination threshold can be reached fast with a fixed time table. We have built our experiments with different big datasets and the results manifest the performance of our design as well as the fast convergence with a limited termination threshold.

The remainder of the paper is organized as follows: Section 2 is the related work; the problem of multilevel association rules mining is formally described in Section 3; in Section 4, the genetic algorithm based multilevel association rules mining is presented; in Section 5, the performance of proposed method is evaluated on several big datasets; the conclusions are drawn in Section 6.

2. State of the Art

Many researchers have focused on multilevel association rules mining. The first branch are Apriori [6] based methods. To mine multilevel association rules, these methods are either adding all the ancestors of frequent items in the corresponding transaction database, for example, cumulate [9], or exhaustively finding all frequent items in every concept level, for example, ML-T2L1 algorithm [10] and the Level-Crossing algorithm [11]. Another branch is FP-growth [7] based methods, such as methods proposed in [12, 13]. Additionally, Cao et al. [12] and Tang et al. [13] expanded the FP-tree with ancestors of items. Wan built an approach through grouping and merging the single level association rules generated by FP-growth. Comparing with Apriori based methods, the FP-tree based methods inherit the merit of FP-tree algorithm which takes less times to scan the dataset and finds the multilevel frequent items. However, when they are utilized to analyze big data, the computational and memory cost will increase exponentially which leads to a prominent bottleneck in big data analysis.

In addition, there are some other approaches to improve the efficiency of multilevel association rules mining. Vejdani et al. proposed a method that extracted multilevel membership functions by Ant Colony Systems algorithm without specifying the actual minimum support [14]. To enhance the efficiency of computing, Mahmoudi et al. optimized Vejdani's method by fixing the functions for each item followed by computing minimum supports [15]. Wang et al. took advantages of the OLAP and data mining technology in multilevel association rules mining which brought efficiency and flexibility [16]. Besides, mining association rules with genetic algorithm (GA) [17] based methods have also been explored. The GA-based methods are able to quickly scan association rule candidate set with large amount of candidates. According to the previous research work [18], the GA-based methods can discover high-level prediction rules. This is because the GA-based methods perform a global search on association rules and can handle the data with attribute interactions better, comparing with the greedy rule induction algorithms. Previous researches have thoroughly explored single-level association rules mining with GA, such as mining single-objective rules [19] and mining multiobjective rules [20]. However, in the big data analysis context, strong association rules are always in multilevel forms, and mining multilevel association rules in big data needs more efficient methods. The GA-based multilevel association rules mining method proposed in this paper is one attempt to efficiently find multilevel association rules in big data.

3. Problem Description

The multilevel association rules mining problem can be described as follows: there are a set of items $I = \{i_1, i_2, \ldots, i_n\}$ and Γ is a catalog tree that briefly defines the multilevel categorizing relationships between items as the domain knowledge. i_1 is a parent of i_2 and i_2 is a child of i_1 if there is an edge in Γ from i_1 to i_2. We denote $\hat{\imath}$ as an ancestor of i and i as a descendant of $\hat{\imath}$ if there is a path from $\hat{\imath}$ to i in Γ. Only leaf nodes are presented in the database. An illustration of a catalog tree in a supermarket domain is shown as in Figure 1.

D is a database of transactions where each transaction T in D is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier *TID*. Items in T are expected to be leaves in Γ. Note that a transaction T *supports* an item $x \in I$ if x is in T or x is an ancestor of some items in T. In addition, a transaction T *supports* $X \subseteq I$ if T supports every item in X.

A multilevel association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \varnothing$. No item in Y is an ancestor of any item in X; that is, $Y \cap \text{ancestors}(x) = \varnothing$. This is because a rule of the form "$x \Rightarrow \text{ancestor}(x)$" is trivially true with 100% confidence, which is redundant. Both X and Y can contain items from any level of Γ.

The rule $X \Rightarrow Y$ holds in transaction set D with *support* s, where s is the percentage of transactions in D that support $X \cup Y$. $X \cup Y$ indicates the probability $P(X \cup Y)$. The rule $X \Rightarrow Y$ has *confidence* c in transaction set D, where c is the percentage of the transactions which support X in D that meanwhile support Y. This can be represented as the conditional probability $P(Y \mid X)$. Then,

$$\begin{gathered}\text{Support}(X \Longrightarrow Y) = P(X \cup Y), \\ \text{Confidence}(X \Longrightarrow Y) = P(Y \mid X).\end{gathered} \tag{1}$$

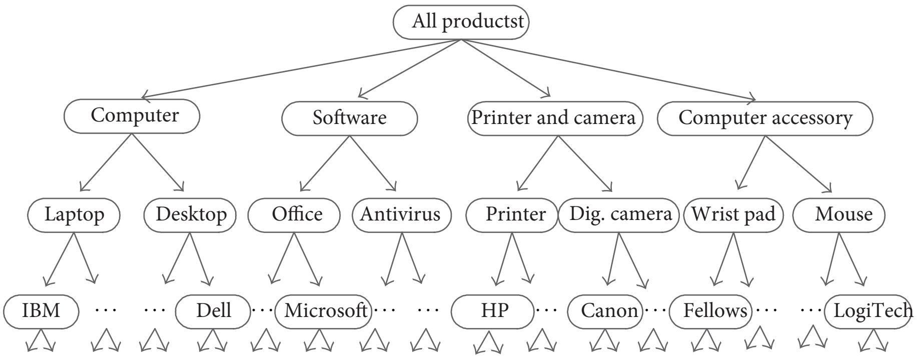


Figure 1: A catalog tree in supermarket domain.

Table 1: Association rule threshold.

Rules	Minsupp	Minconf
$Computer \Rightarrow Office$	5%	50%
HPprinter ⇒ *Canon Camera*	1%	50%

Table 2: Valid association rules.

Rules	Support	Confidence
$Computer \Rightarrow Office$	10%	66%
HPprinter ⇒ *Canon Camera*	0.8%	66%

Example. Let Γ be the catalog tree shown in Figure 1. The *Minsupp* and *Minconf* are shown in Table 1, and two of the rules on item sets are shown in Table 2. Note that the rule "Computer ⇒ Office" satisfies the minimum support (5%) and the minimum confidence (50%), but the rule "HP printer ⇒ Canon Camera" does not satisfy the minimum support (1%). Therefore, rule "Computer ⇒ Office" is considered as a valid multilevel association rule.

To evaluate the rules discovered from the multilevel abstraction, we prefer the following.

(i) Support Confidence. The rules with larger support and higher confidence are preferred, where larger support reflects that the rules are more general, and higher confidence reflects the certainty of discovered rules in the domain statistics.

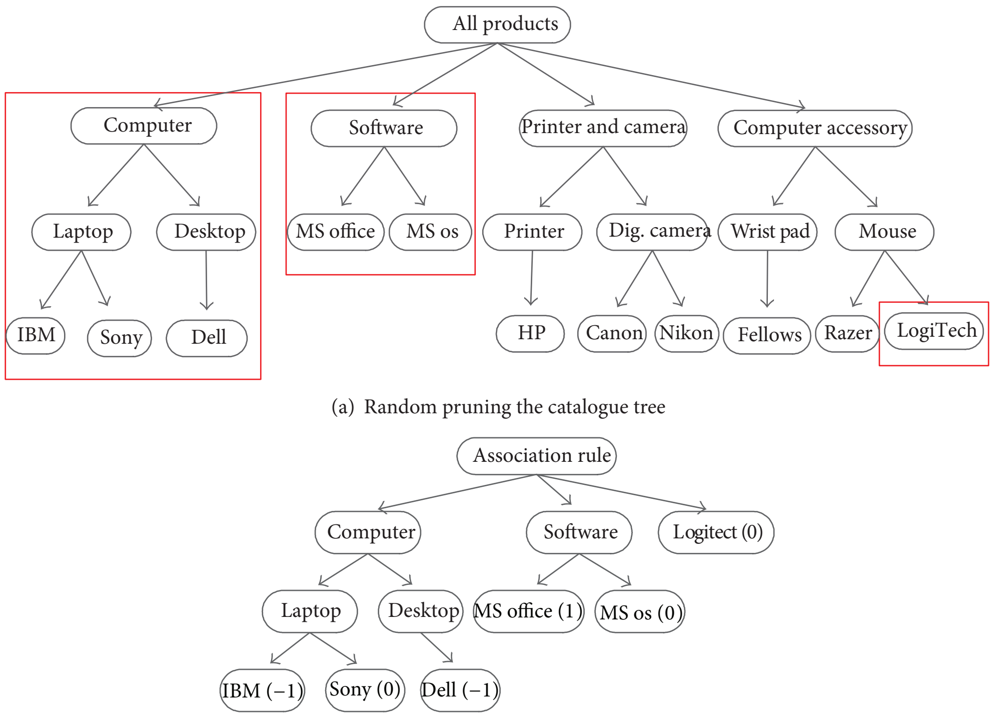
(ii) Interest. The rule in a proper level of catalog tree is preferred. Mining association rules at low levels may lead to uninteresting rules that are too trivial; that is, "IBM_ThinkPaD_R40/P4M ⇒ Symantec_Norton_Antivirus_2003." However, mining association rules at high levels usually leads to common sense, for example, "Computer ⇒ Software."

Mining of multilevel associations involves items at different levels of abstractions and its exhaustive computation complexity has been classified as NEXP-COMPLETE. A dataset that contains k items in the primitive level can potentially generate up to $2^k - 1$ primitive and nonempty frequent item sets. Particularly, in the *Bigdata* context, with the number of items in the catalogue and transactions increasing rapidly, the computational and memory consumptions of the traditional methods will be expanded exponentially. It is worthy of noting that the FP-tree algorithm enhances the efficiency of mining association rules, but it can hardly mine the multilevel association rules, especially the cross-level association rules. Thus, in big data analysis context, a novel heuristic method is imperative to mine multilevel association rules.

4. GA-Based Approach

Genetic algorithm is a heuristic search approach that mimics the process of natural evolution and generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Its essence is an efficient, parallel, and global search method, which can automatically obtain and accumulate knowledge about the search space, and control the search space in order to achieve the optimal solution adaptively in the search process. In the traditional multilevel association rules mining algorithms, we have to generate almost all candidate items and test them against the entire database. However most of the mining process is in vain and leads to heavy computational cost. The genetic algorithm offers a novel way to solve these problems. By efficiently testing the most likely candidate items preferentially, GA-based method can control the search space and achieve the optimal solution adaptively during the association rules searching. Therefore, by taking this advantage, the association rules search space will be greatly reduced and the performance of mining method can be dramatically improved.

4.1. Encoding Scheme. When GA is applied to mine multilevel association rules, a key is to encode and automatically generate candidates of the association rules in a GA-based form. Because the classic GA-based encoding schema is not feasible to mine multilevel association rules, we propose a new category tree based encoding scheme to represent the association rule candidates. Each valid multilevel association rule can be modeled as a subtree of the catalog tree, but each leaf node should be assigned with a binary number to indicate whether it is an antecedent or a consequent in the rule. The goal of the algorithm is to find valid candidates



(a) Random pruning the catalogue tree

(b) An association rule

Figure 2: Illustration of the encoding schema.

by the evolution of subtrees. The structure of the GA-based encoding subtree can defined as follows:

$$\begin{aligned}\text{Class TreeNode}\,\{&\text{int Id;}\\ &\text{string attributeName;}\\ &\text{int assignedValue;}\\ &\text{TreeNode parentNode;}\\ &\text{List}\,\langle\text{TreeNode}\rangle\,\text{childList;}\}\,.\end{aligned}\tag{2}$$

In this representation, every leaf represents a commodity and is assigned to a value of 0, 1, or –1 as shown in Figure 2. The antecedents of the association rule can be expressed by the commodities assigned to 0 and the consequents can be expressed by the ones assigned to 1. The commodities assigned to –1 do not join the association rules. With randomly pruning the catalogue tree and assigning the values of the leaves, we can initiate the subtrees with the first generation of the association rules. For example, the items in red rectangle of Figure 2(a) are regarded as the children of the association rules tree. The multilevel association rules can be generated and polished by the processes of roll-up, mutation, crossover, and selection operator.

Example. As shown in Figure 2(b), the leaves assigned to 0 represent the antecedents of the association rule and the leaves assigned to 1 represent the consequents, so the association rule can be represented as (Sony ∩ MS os ∩ Logitech) ⇒ (MS office). The rule implies that people who buy Sony laptops and Logitech mouses, MS os, will be likely to buy MS office software.

4.2. Genetic Operators. Initially, according to the given catalog tree, we will randomly prune the catalog tree to get the subtrees as the children of the association rules trees. Then we randomly assign the leaves of each association rule tree to –1, 0, or 1 and make sure that the association rule tree maintains 0 and 1 at the same time. In the same way, we can get the appropriate number of the initial population.

Selection operator defines how to choose the individuals that will create the offsprings for the next generation. The selection operator is based on the fitness function that the offsprings with high fitness will have higher probabilities to be selected. In this paper, we use "roulette wheel" [21] selection, and the higher the fitness of an individual is, the more likely it is to be selected to reproduce.

After the high fitness individuals are selected, the crossover operator can be applied. This function allows a pair of selected subtrees to cross over at one of their randomly chosen common nodes to produce new generations so as to avoid generating invalid rules. In particular, only attribute values will be exchanged if only leaf nodes are crossed over. The crossover of the root node is prohibited because no new rules will be produced. An example of crossover process in a real domain is illustrated as in Figure 3.

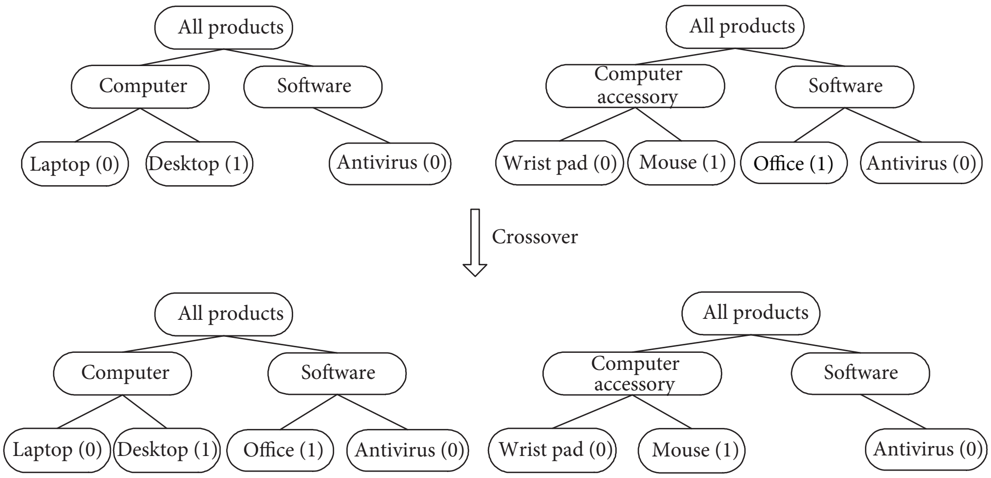
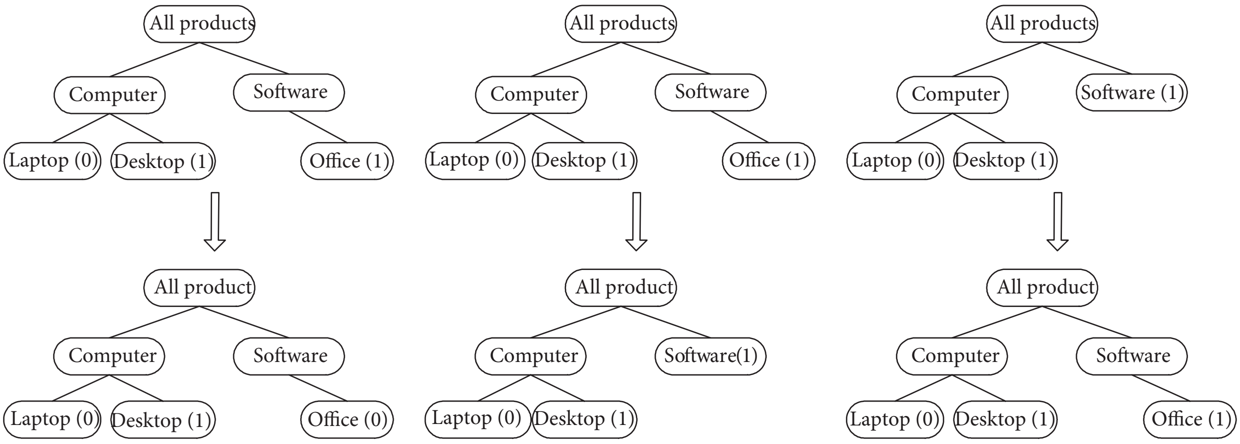


Figure 3: Illustration of the crossover.



(a) Alter attribute value

(b) Prune subtree

(c) Add subtree

Figure 4: Illustration of the mutation.

Mutation operator plays an important role in maintaining the diversity of the population during the mutations. In our schema, we define three types of mutation operators:

(1) randomly choose a leaf node and assign with an alternative attribute value;

(2) randomly choose a nonroot node and prune its subtree;

(3) randomly choose a nonroot node and add a subtree to the node.

An illustration of the mutation operators is shown as in Figure 4, where the node with thick border indicates that it is the one to be chosen and mutated.

4.3. Fitness Function. The fitness function plays an important role in our GA schema. It is used either to evaluate the offsprings that will be selected into the next generation or to act as the terminate condition, when there have been enough association rules with higher fitness values that are more than the predefined threshold. To build the fitness function, we have to combine the *support* and *confidence* attributes, which are necessary to described an association rule in domain, in our fitness definition. Therefore, the fitness of an association rule $A \Rightarrow B$ is defined as

$$F(A \Longrightarrow B) = \alpha \times \text{support}(A \Longrightarrow B) + \beta \times \text{confidence}(A \Longrightarrow B). \tag{3}$$

Parameters α and β are the important factors to balance the weight of the *support* and *confidence* in the fitness function, and $\alpha + \beta = 1$. To mine valid association rules from the big data base with our GA approach, the threshold of the fitness function has to be predefined. As the threshold is relevant to the *support* and *confidence* attributes, we should set the thresholds of minimum support *min_sup* and the minimum confidence *min_conf* for the algorithm. In our approach, other than uniformly using the same thresholds for all levels, we use different *min_sup* for different levels of association rules. The deeper the level is, the smaller the corresponding thresholds will be. Furthermore, the more leaf

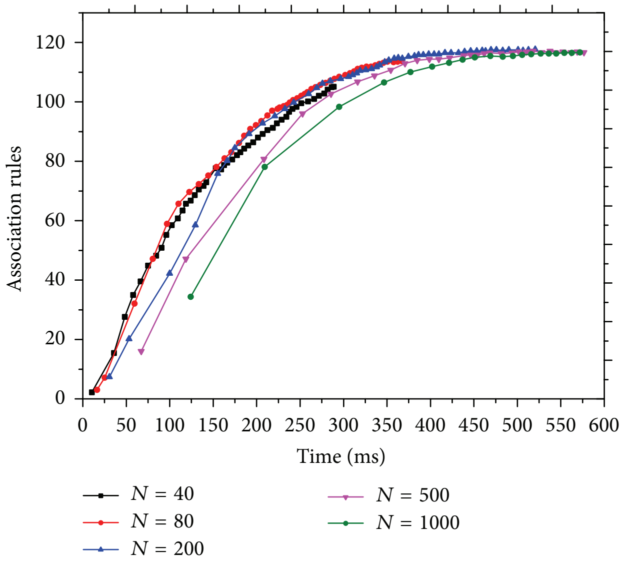


Figure 5: With a limited population and a limited time, most valid association rules could be mined in dataset 1.

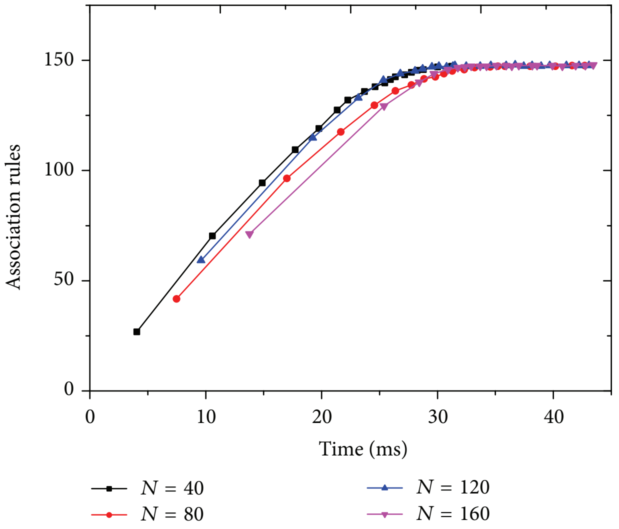


Figure 6: With a limited population and a limited time, most valid association rules could be mined in dataset 2.

nodes the ancestor has, the higher *min_sup* of the ancestor will be [13].

5. Experimental Results

In this section, we build various experiments to analyse the performances of our design. We briefly use two different transaction databases to mine the multilevel association rules: “Dataset 1” from University of Regina (http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/datasets.php/) and “Dataset 2” from California State University Los Angeles (http://www.calstatela.edu/centers/hipic/contents/researchy/cloudComputing/files/market/).

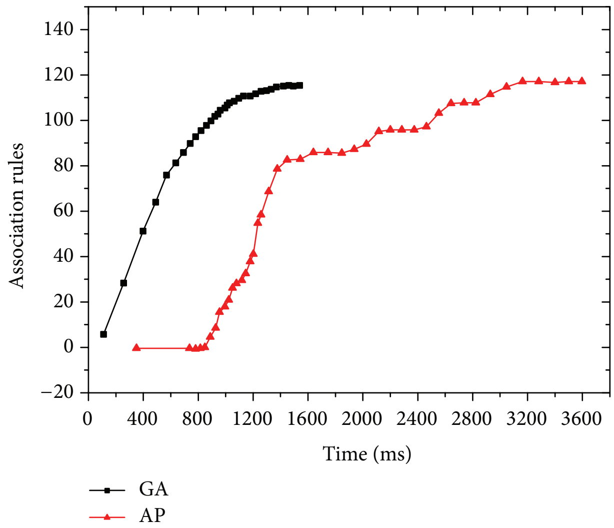


Figure 7: Number of association rules mined from 4000 transactions in dataset 1.

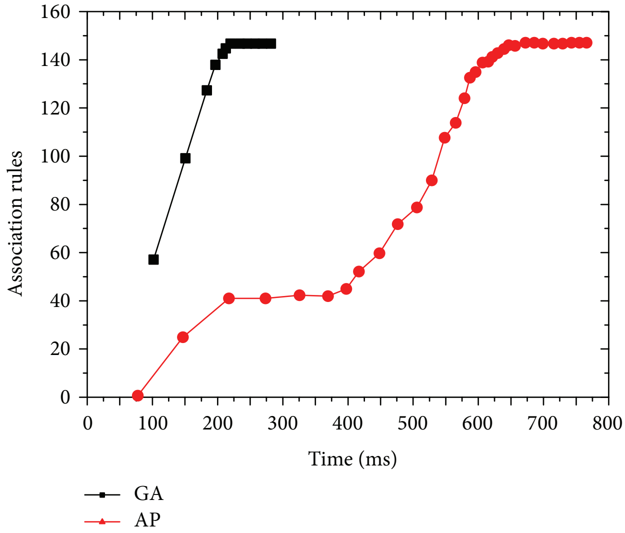


Figure 8: Number of association rules mined from 4000 transactions in dataset 2.

We also use the classic Apriori algorithm as the benchmark to compare with our GA-based algorithm. All the experiments run on a PC with Core i5 CPU and 4 GB RAM. For the default settings, we set the size of the population in our GA-based algorithm for dataset 1 as 200 while for dataset 2 as 200. The initial *min_sup* is 0.01 and *min_conf* is 0.5.

In the first experiment, we test whether valid association rules can be mined with a fixed number of initial generations and in a limited of period. In dataset 1, the initial generation size varies from 40 to 1000, and, in dataset 2, it varies from 50 to 160. The results are shown in Figures 5 and 6. We can conclude that if the population is too small, the performance of the GA-based algorithm will be similar to the random algorithm. But if the population is too large, although we can obtain enough association rules quickly, the

Table 3: Number of association rules mined from 1000 and 2000 transactions in two datasets.

		Results from dataset 1						
Number of transactions	Type	Time (s)						
		62.5	125	250	400	1000	2000	2500
1000	GA	32.8	67.7	94.4	97	97	97	97
	AP	0	0	0	1	56	98	98
2000	GA	7	30.4	66.9	92.3	116.6	116.6	116.6
	AP	0	0	0	0	35	97	118
		Results from dataset 2						
Number of transactions	Type	Time (s)						
		15	20	40	100	200	400	600
500	GA	57.6	97.2	150.9	151	151	151	151
	AP	0	0	0	46.4	130	151	151
1000	GA	0	57.3	130.9	147	147	147	147
	AP	0	0	0	39	74	141	147

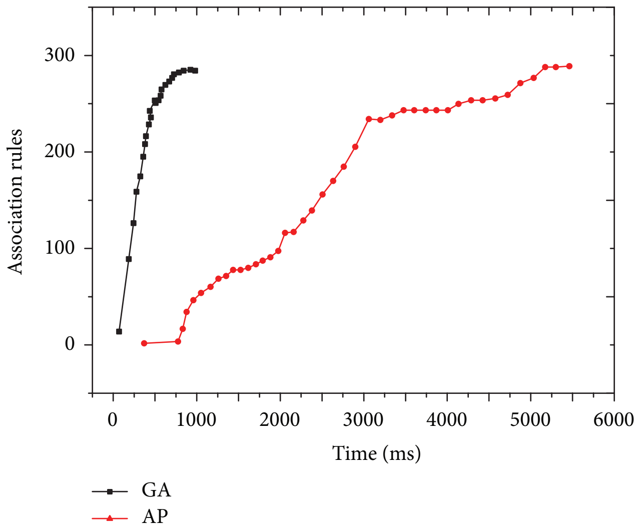


Figure 9: Mining efficiencies of GA-based and Apriori algorithm with different *min_sup* and *min_conf* in dataset 1.

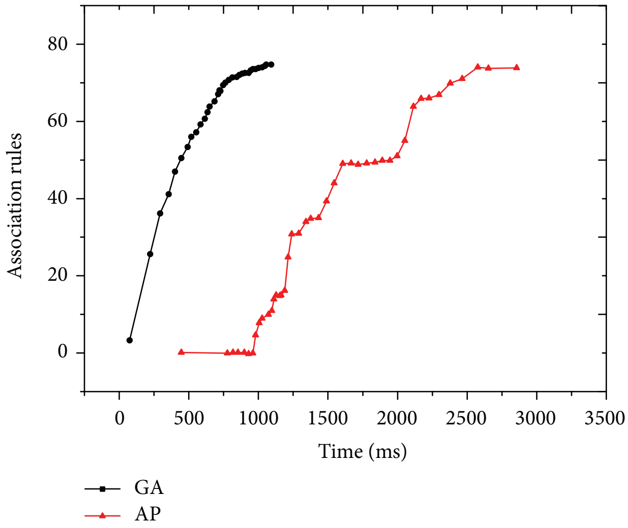


Figure 10: Mining efficiencies of GA-based and Apriori algorithm with different *min_sup* and *min_conf* in dataset 2.

computational complexity grows fast. However, as we can see, in both datasets, there is a good balance that, with a limited population and a limited time period, most valid association rules have been mined. Therefore we select 200 and 120 as the default population for dataset 1 and dataset 2, respectively, which performs well in our algorithm.

In the rest of this section, we compare the efficiency of our GA-based algorithm with the Apriori algorithm in the two different datasets. In Figures 7 and 8, we fixed 4000 transactions in each dataset. As it has shown, with the progress of the time step, GA-based approach can find valid association rules much quicker in both datasets. Although as an exhaustive approach, Apriori algorithm may be able to find a few more valid rules than ours if there is no time limit. However, in most big data applications, the system response time is a critical criterion of its performance and our approach is more valuable to obtain most association rules in a short period of time. In Table 3, we presented more detailed results when we picked 1000 and 2000 transaction records from each dataset. As the time goes by, we reach a consistent conclusion that our GA-based approach is more capable of finding valid association rules than the exhaustive approach.

When we changed the thresholds of the minimal support and confidence for the valid association rules, the output for both GA-based algorithm and the Apriori algorithm is changed. As shown in Figure 9, in dataset 1, we reduced the *min_sup* to 0.005 and the *min_conf* to 0.25. In Figure 10, we increased the *min_sup* to 0.02 and the *min_conf* to 0.75 for dataset 2. In both experiments, the GA-based algorithm finds more association rules in a short of period than the Apriori algorithm, no matter what the multilevel association thresholds are.

In the next group of experiments, we tested how valuable the mined association rules are from either the GA-based algorithm or the Apriori. To measure its value, we use the same formula of fitness function and set $\alpha = 0.9$, $\beta = 0.1$.

Table 4: Average fitness of association rules mined from 1000 and 2000 transactions in two datasets.

		Results from dataset 1							
Number of transactions	Type	Time (s)							
		20	62.5	125	250	400	1000	2000	2500
1000	GA	0.14	0.17	0.17	0.16	0.16	20	62.5	125
	AP	0	0	0	0	0.13	0.15	0.16	0.16
2000	GA	0	0.21	0.24	0.23	0.22	0.20	0.20	0.20
	AP	0	0	0	0	0	0.16	0.20	0.20
		Results from dataset 2							
Number of transactions	Type	Time (s)							
		15	20	40	100	200	400	600	1000
500	GA	0.35	0.36	0.35	0.35	0.35	0.35	0.35	0.35
	AP	0	0	0	0.29	0.36	0.35	0.35	0.35
1000	GA	0	0.36	0.37	0.35	0.35	0.35	0.35	0.35
	AP	0	0	0	0.26	0.28	0.36	0.35	0.35

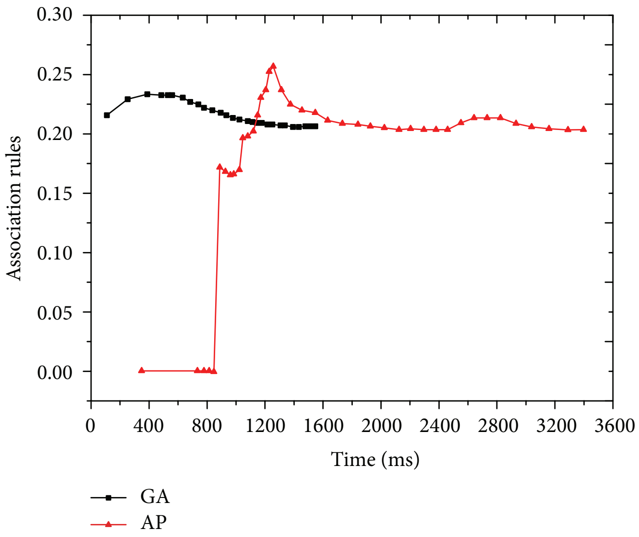


Figure 11: Average fitness of association rules mined from 4000 transactions in dataset 1.

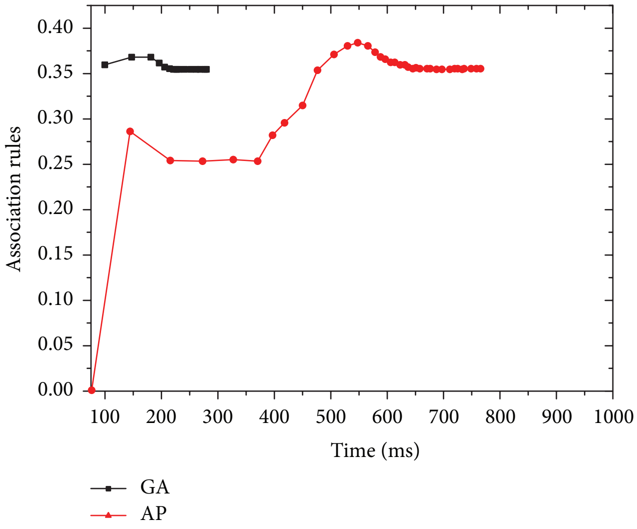


Figure 12: Average fitness of association rules mined from 4000 transactions in dataset 2.

The experiment results in different datasets with the 4000 transactions are shown in Figures 11 and 12, and the results in datasets with different number of transactions are shown in Table 4.

Consistent with the conclusions above, with the progress of the time horizon, GA-based approach can find high valuable association rules much quicker than Apriori in both datasets.

6. Conclusion and Future Work

In this paper, we have presented a novel genetic based algorithm to mine multilevel association rules in big date sets. By utilizing the application domain knowledge that could be briefly explained as a catalog tree, we introduce a special subtree based encoding schema to make the GA-based algorithm possible. In addition, we personalized the initiation, crossover and mutation functions for the tree-based genetic operators. Based on our simulations and the experiment results, we can see that by building the dynamic fitness function from the multilevel support and confidence threshold, this algorithm is adaptive and convergent. Moreover, we test its performance in different databases and the algorithm performs better than the classic Apriori algorithm with faster and more accurate mining the high quality multilevel association rules.

Although we have proved that our GA-based approach is capable of dealing with some key challenges in the multilevel association rule mining in big databases, we leave many of the others in the future. Firstly, our approach is only good in the domains that the items in the association rules can be organized as a catalog tree. But when it is applied in some other domain with unstructured item sets, our basic design does not match. Second, our GA-based approach should be encoded in a distributed and parallel computation environment so as to optimize its performance. Moreover, deployment in real domain is the key to evaluate our approach and polish the algorithm for better performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research has been sponsored in part by National Natural Science Foundation of China nos. 61370151 and 61202211, National Science and Technology Support Program of China 2012BAI22B05, and Huawei Research Foundation YB2013120141.

References

[1] E. Dumbill, "Making sense of big data," *Big Data*, vol. 1, no. 1, pp. 1–2, 2013.

[2] K. Cukier and V. Mayer-Schoenberger, "The rise of big data: how it's changing the way we think about the world," *Foreign Affairs*, vol. 92, no. 3, 2013.

[3] B. Petelin, I. Kononenko, V. Malačič, and M. Kukar, "Multi-level association rules and directed graphs for spatial data analysis," *Expert Systems with Applications*, vol. 40, no. 12, pp. 4957–4970, 2013.

[4] B. Fan and J. Luo, "Spatially enabled emergency event analysis using a multi-level association rule mining method," *Natural Hazards*, vol. 67, no. 2, pp. 239–260, 2013.

[5] D. Han, Y. Shi, W. Wang et al., "Research on multi-level association rules based on geosciences data," *Journal of Software*, vol. 8, no. 12, pp. 3269–3276, 2013.

[6] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference: Very Large Data Bases*, vol. 1215, pp. 487–499, 1994.

[7] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD Record*, vol. 29, no. 2, pp. 1–12, 2000.

[8] G. Alterovitz, M. Xiang, M. Mohan, and M. F. Ramoni, "GO PaD: the gene ontology partition database," *Nucleic Acids Research*, vol. 35, supplement 1, pp. D322–D327, 2007.

[9] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," in *VLDB*, vol. 95, pp. 407–419, 1995.

[10] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in *Proceedings of the 21st International Conference: Very Large Data Bases*, vol. 95, pp. 420–431, 1995.

[11] R. S. Thakur, R. C. Jain, and K. R. Pardasani, "Mining level-crossing association rules from large databases," *Journal of Computer Science*, vol. 2, no. 1, 2006.

[12] H. Cao, Z. Jiang, and Z. Sun, "Fast mining algorithm for multi-level association rules based on FP-tree," *Computer Engineering*, vol. 19, no. 25, 2007.

[13] H. Tang, M. Wu, and Y. He, "Improved multilevel association rule mining algorithm," *Computer Engineering*, vol. 16, no. 16, 2011.

[14] E. Vejdani, F. Saadatmand, M. Niazi, and M. H. Yaghmaee, "Extracting membership functions by ACS algorithm without specifying actual minimum support," in *Proceeding of the International Conference on Multimedia Computing and Information Technology (MCIT '10)*, pp. 13–16, Sharjah, United Arab Emirates, March 2010.

[15] E. V. Mahmoudi, E. Sabetnia, M. N. Torshiz, M. Jalali, and G. T. Tabrizi, "Multi-level fuzzy association rules mining via determining minimum supports and membership functions," in *Proceedings of the 2nd International Conference on Intelligent Systems, Modelling and Simulation (ISMS '11)*, pp. 55–61, Kuala Lumpur, Malaysia, January 2011.

[16] Y. Wang, L. Yu, Q. Wen et al., "Improved multi-level association rule in mining algorithm based on a multidimensional data cube," in *Proceedings of the Consumer Electronics, Communications and Networks (CECNet '13)*, pp. 355–358, 2013.

[17] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.

[18] A. A. Freitas, *A Survey of Evolutionary Algorithm s for Data Mining and Knowledge Discovery*, Advances in Evolutionary Computing Natural Computing Series, 2003.

[19] M. R. Kumar and D. K. Iyakutti, "Application of genetic algorithms for the prioritization of association rules," *IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches and Practical Applications*, pp. 1–3, 2011.

[20] M. K. Gupta and G. Sikka, "Association rules extraction using multi-objective feature of genetic algorithm," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, San Francisco, Calif, USA, October 2013.

[21] C. R. Houck, J. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a matlab implementation," *NCSU-IE TR*, vol. 95, no. 9, 1995.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization