Data Association Rules Mining Method Based on Improved Apriori Algorithm

Haotong Wu*
University of California San Diego, USA

ABSTRACT

In the existing data mining technology, there are some shortcomings in association rule mining methods. In this paper, aiming at the problem that the mining efficiency of Apriori algorithm is not high when dealing with large database, the genetic algorithm is introduced to improve the Apriori algorithm. This paper first introduces the basic concept and principle of association rules, describes the basic idea of Apriori algorithm in detail, and an improved algorithm based on Partition algorithm is proposed for its shortcomings. Then introducing the principle and operation flow of genetic algorithm in detail, and puts forward the corresponding improvement solutions for the coding scheme and fitness function. Finally, the association rule mining is established by combining genetic algorithm with Apriori algorithm. Compared with other methods, the experimental results show that the proposed method has better performance, higher mining efficiency and better data mining.

CCS CONCEPTS

• **Theory of computation** → Theory and algorithms for application domains; Algorithmic game theory and mechanism design; Algorithmic mechanism design.

KEYWORDS

Apriori algorithm, genetic algorithm, coding scheme, fitness function, association rules, Partition algorithm

ACM Reference Format:
Haotong Wu*. 2020. Data Association Rules Mining Method Based on Improved Apriori Algorithm. In *2020 the 4th International Conference on Big Data Research (ICBDR'20) (ICBDR'20), November 27–29, 2020, Tokyo, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3445945.3445948

1 INTRODUCTION

With the progress of science and technology and the continuous update of Internet technology, a large number of data have been produced in various fields of society. In order to get hidden and useful information from massive data, data mining technology emerges as the times require. Data mining, in short, is to use a variety of specific algorithms and methods to mine valuable information from the data. Association rule analysis is the most active branch, especially worthy of in-depth study. In recent years, although some research results have been achieved, it still has certain research value. However, although the association rule mining algorithm has made some achievements, the amount of data is increasing, which leads to some shortcomings and deficiencies of the existing association rule mining algorithm, which seriously affects the quality of association rule mining. Because the amount of data in the database is increasing every day, and each record has hundreds of attributes, plus some Because of the Partition of continuous attributes, the data of itemset is very large. The classical Apriori algorithm takes the whole database as the solution space, and needs to scan the database frequently, which makes the efficiency of the whole algorithm decrease. Therefore, how to improve the Apriori algorithm to meet the real needs of data mining in the era of big data and improve the efficiency of the algorithm has become the top priority of researchers.

ICBDR'20, November 27–29, 2020, Tokyo, Japan
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8775-0/20/11…$15.00
https://doi.org/10.1145/3445945.3445948

2 THECONCEPT AND PRINCIPLE OF ASSOCIATION RULES

Association rules analysis is to search for repeated relationships or relationships from transaction data sets, and the discovered relationships are represented by association rules or frequent item sets. The basic definition of association rules is as follows:

Definition 1: the data set of rule mining is marked as D = $\{d_1, d_2, \cdots, d_k, \cdots, d_n\}$,$s_k = \{i_1, i_2, \cdots, i_m, \cdots, i_p\}$, $d_k(k = 1, 2, \cdots, n)$is transaction, $i_m(m = 1, 2, \cdot\cdot\cdot, p)$ is item. Each record is identified by a unique TID identifier.

Definition 2: assumes that association rules are in the form of $X \rightarrow Y$, where the relationship between item set X and item set Y is $X \cap Y = \emptyset$. The quality of association rules can be measured by support S and confidence C. The support degree represents the probability that X and Y appear simultaneously in the whole transaction, which represents the importance of the rule, while the confidence can be expressed by the probability of Y appearing in the transaction containing X, that is, $P(X|Y)$ conditional probability, which represents the credibility of the rule. Its formal definition is as follows:

$$S(X \rightarrow Y) = S(X\text{U}Y) = P(X\text{U}Y) \tag{1}$$

$$C(X \rightarrow Y) = \frac{S(X \cup Y)}{S(X)} = \frac{P(X \cup Y)}{P(X)} \tag{2}$$

Definition 3: A set of items is called a item set, and a set containing k items is called a k item set. If the support of item set I is greater than the minimum support count, I is called frequent item set, and frequent k item set is generally represented by L_k.

Definition 4: If the rule $S(X \rightarrow Y) \geq min\_sup$ and $C(X \rightarrow Y) \geq min\_conf$, where min_sup is the minimum support threshold, min_conf is the minimum confidence threshold, the association rule $X \rightarrow Y$ is called strong association rule. The task of association rule is to mine the strong association rule in data set D.

The process of mining association rules is generally divided into two steps, as follows:

(1) According to the definition of frequent itemsets, the frequency of each itemset is at least the same as the preset minimum support count.

(2) According to the definition of strong association rules, the generated rules need to meet the minimum support and minimum confidence threshold at the same time.

3 APRIORI ALGORITHM AND ITS EVALUATION

3.1 Overview of Apriori algorithm

Apriori algorithm is an algorithm for mining frequent item sets of Boolean association rules. The algorithm is named as we can see. It is based on prior knowledge to mine frequent item sets. The iterative method of layer by layer search is used. At first, according to the minimum support degree, the frequent 1 item set is recorded as L_1, and then it is pushed forward layer by layer. The frequent 2-item set L_2 is generated by frequent 1-item set L1. Finally, frequent k+1 item set L_{k+1} is generated by frequent k-item set L_k, until frequency can no longer be found Item set L_k. Apriori algorithm is composed of join step and pruning step. In order to improve the efficiency of generating frequent item sets, the apriori principle is: if an item set is frequent, then all its non empty subsets are also frequent.

3.2 The basic idea of Apriori algorithm

The process of Apriori algorithm is a process of finding frequent item sets from candidate sets. The flow chart of the algorithm is shown in Figure 1

The implementation steps of Apriori algorithm are as follows

(1) For the whole transaction database, the minimum support count threshold is defined, and k is taken as the initial value of frequent K item set, which is recorded as 1;

(2) Scanning the whole transaction database and calculate the support count of each item set;

(3) The support count of each set is compared with the minimum support threshold. If the support count is greater than the minimum support threshold, it is retained. Otherwise, the frequent k item set is obtained;

(4) Judging whether the generated frequent k item set is empty. If it is empty, the algorithm will end and get the final frequent k-1 item set $L_{k\text{-}1}$. Otherwise, the frequent k item set L_k itself is connected to generate the candidate k+1 item set G_{k+1}. ;

(5) Scanning the database and pruning the candidate set according to a priori p.rinciple, that is to delete the subset of candidate item set. If the item set is not a frequent item set, then calculate the support count of each item set of candidate G_{k+1}, delete the item set that is less than the minimum support count threshold, and finally judge whether the k+1 item set is empty. If it is, the algorithm ends; otherwise, execute step 6;

(6) Let k= k+1, go to step 2, repeat all operations until the k item set is empty, and the whole algorithm runs to get the final frequent item set.

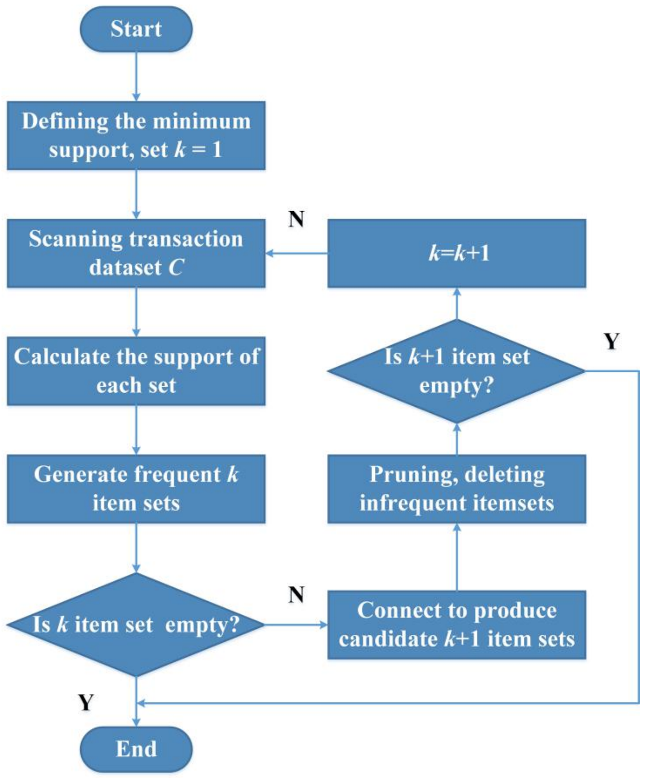
Start
Defining the minimum support, set k = 1
Scanning transaction dataset C
N
k=k+1
Calculate the support of each set
Is k+1 item set empty?
Y
Generate frequent k item sets
Pruning, deleting infrequent itemsets
N
Is k item set empty?
Connect to produce candidate k+1 item sets
Y
End

Figure 1: Flow chart of Apriori algorithm

3.3 Evaluation of Apriori algorithm

Although Apriori algorithm is very simple, there is no complex theoretical formula to assist reasoning, and it is also very easy to implement, but it still has some unavoidable defects

(1) Frequent scanning of transaction databases is required. In the description of Apriori algorithm, it is necessary to scan the whole database for many times, so as to calculate the support count of item set. The overhead is too large. Especially in the transaction database with large amount of data, the time spent in scanning the database once exceeds the acceptable range. At the same time, due to the limitation of memory space, the Apriori algorithm will also lead to the In the actual implementation process, the efficiency is very low;

(2) It produces a very large set of candidates. Since the self join operation of frequent item sets L_k is used to generate candidate item sets, the number of candidate item sets generated will increase exponentially with the increase of k. with the increase of rule length, the number of candidate item sets generated will cause great trouble to the operation

3.4 Improvement of Apriori algorithm

In order to reduce the computer consumption and improve the performance of the algorithm, this paper designs an improved Apriori algorithm based on Partition algorithm. Partition algorithm needs to scan the whole database twice to obtain frequent item sets. To be specific, firstly, the whole transaction database is divided into n non overlapping sub Partitions. The Partition standard is that the data of each Partition can be put into memory at one time. Then, all frequent item sets are found for each Partition. After all, the frequent item sets obtained by Partitioning do not represent the frequent item sets of the whole data set, but the frequent item sets of the whole transaction data set must be Finally, the whole database needs to be scanned to calculate the true support count of each Partition's frequent item sets, so as to determine the overall frequent item sets.

The core part of the algorithm is divided into two steps.

Firstly, the database C used to extract association rules is divided into disjoint databases $D^i(i = 1, 2, \cdots, n)$. To control the size of the database, ensure that each sub transaction database can be put into memory. The sub transaction database is $D^i(i = 1, 2, \cdots, n)$. Then the Apriori algorithm is used to find the strong point set L^i in the database D^i, and then all the strong points are set into the potential strong points set in the database D:

$$l_p = \cup_{i=1}^{n} L^i \tag{3}$$

Secondly, after the initialization of potential items, genetic algorithm is used to get the strong item set of latent item set l_p. In this process, the database only needs to be accessed twice, which reduces the pressure on the database and improves the performance of the algorithm.

4 APRIORI-GA HYBRID MODEL

4.1 Genetic Algorithm

Genetic algorithm (GA) is a kind of bionic evolutionary algorithm, which comes from the theory of biological evolution. It transforms the problem to be solved into the process of biological evolution, and generates the next generation through coding, selection, crossover, mutation and other operations. At the same time, the individuals with lower fitness function will be eliminated. Thus, through continuous iterative evolution, individuals with higher fitness value will be produced, and thus the individual with higher fitness value will be generated through continuous iterative evolution and the solution satisfying the condition is obtained. The flow chart of genetic algorithm is shown in Figure 2

The basic operation steps of genetic algorithm are as follows:

Step 1: Coding. When using genetic algorithm to solve the problem, we need to consider the problem of coding. Generally, binary coding is used. The problem to be studied is transformed into binary string, which is easy to operate;

Step 2: Population initialization. A certain number of populations are randomly generated as a set of solutions of the problem to be optimized, in which each individual is composed of a certain length of binary strings;

Step 3: Calculating individual fitness value. The fitness value of each individual population needs to be calculated by the fitness function, and the advantages and disadvantages of the individual population can be judged by the fitness value.

Step 4: Selection operation. The so-called choice is to select the individuals with high fitness value from a group of individuals, that is, the survival of the fittest and the selection of the best;

Step 5: Crossover operation. Two individuals are randomly selected from the population, and the single point crossover operation is performed at different positions according to the set crossover probability;

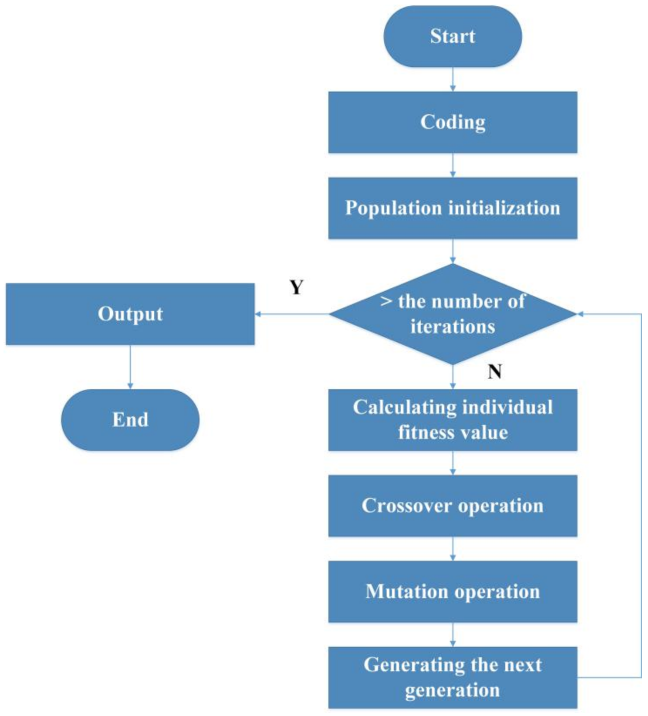
Start
Coding
Population initialization
> the number of iterations
Y
N
Output
End
Calculating individual fitness value
Crossover operation
Mutation operation
Generating the next generation

Figure 2: Flow chart of genetic algorithm

Step 6: Mutation operation. The advantage of mutation is to increase the global optimization of the algorithm;

Step 7: after a series of selection, crossover, mutation and other operations to generate the next generation population, judge whether the generated individuals meet the optimality or the number of iterations. If the requirements are met, the algorithm runs to the end. Otherwise, it turns back to step 3 to continue the cycle operation until the termination conditions are met.

4.2 Design of Model

4.2.1 Coding Strategy. In order to solve the problem of complex parameters, this paper adopts multi-parameter coding technology. In other words, each parameter is encoded to obtain a string group, and then the strings are combined into a whole chromosome. Because each chromosome presents an association rule, which is the goal of data mining. In addition, binary coding method is used.

4.2.2 Design of Fitness Function. Support and confidence are two main aspects of evaluating association rules. When mining association rules, the biggest problem is how to define the association rules of min_sup and min_conf. In this case, the fitness function is defined as:

$$\mathrm{Fit}(x) = a * S(x) + b * C(x) \tag{4}$$

In this function, x is a regular variable, while a and b are constants, which are the shares of support and credibility in the evaluation, and satisfy the requirements of $0 \le a, b \le 1$. When an individual cannot be explained by a reasonable rule, we consider the individual to be useless. We can set $Fit = 0$, that is, the fitness value is zero.

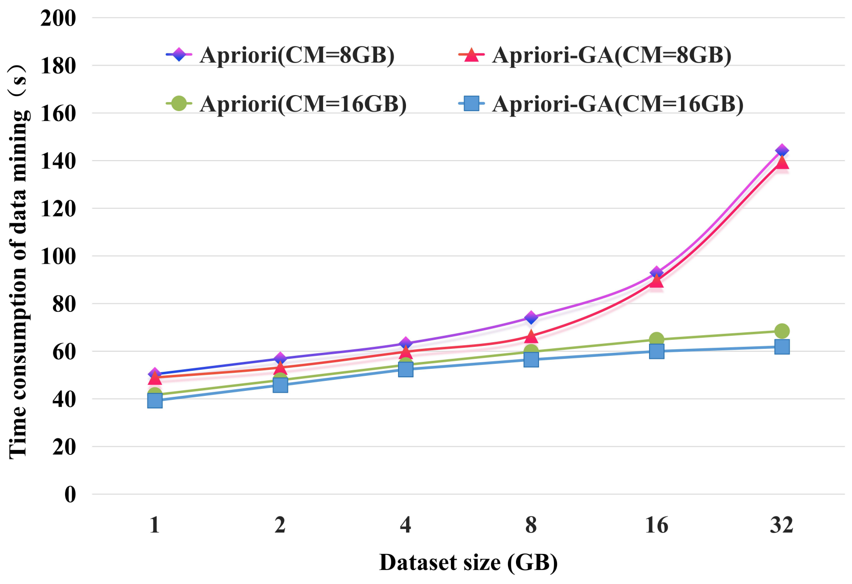

Figure 3: Performance comparison of the 2 algorithms

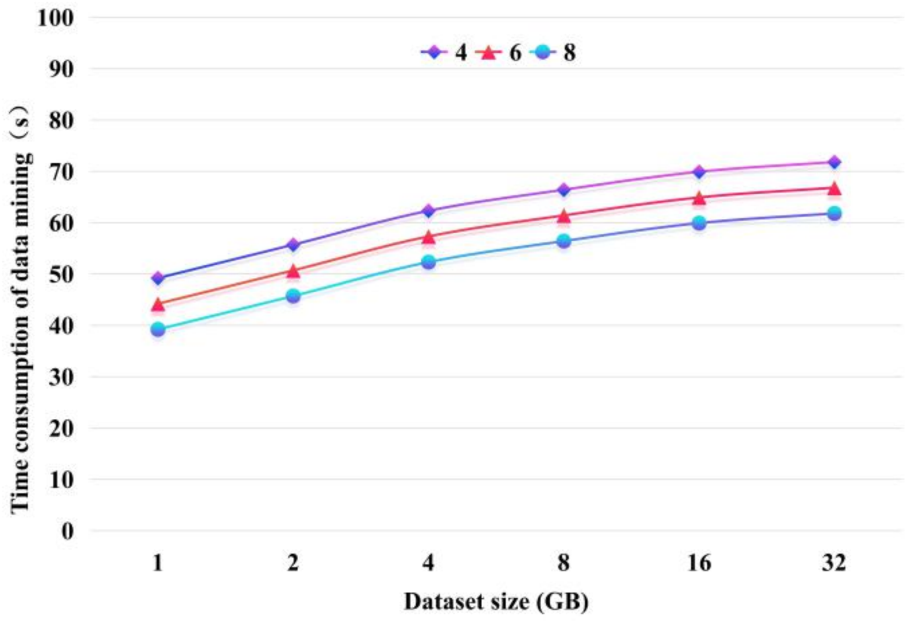

Figure 4: Influence of partition times on algorithm performance

The value of variables *a* and *b* need to be adjusted according to the actual situation of the user

4.2.3 Design of Genetic Operation. (1) Selection operator

In the simple mode of genetic algorithm, the mating group selection problem can take the method of gambling. This method is easy to lead to the search process with the optimal solution, even if it is improved or to obtain the maximum individual competitiveness, unable to complete the principle of the algorithm. Therefore, it is solved by adding sorting algorithm, that is, sorting function of individual fitness value in population is realized by sorting algorithm.

(2) Crossover operator

In the design of crossover operator, the following two problems should be considered: the determination of crossover point and the exchange of genes. Single point crossover is the simplest mode in the crossover operator, because it is in the form of binary coding, this crossover mode has little impact on the performance of the whole algorithm, and is better than multi-point crossover in performance, so this paper uses single point crossover.

(3) Mutation operator

In this paper, we need to determine the location of mutation points and gene replacement when designing mutation factors. In this paper, we use the uniform mutation method, which uses random numbers in a range to calculate the value of a random change locus with a small probability. This method can not guarantee the integrity of gene attributes. In this paper, we will use a dynamic probability P_m to carry out random mutation of genes. After determining the mutation individuals, we will change all the gene loci in turn to ensure that the attribute values on the gene will exist after the mutation is completed.

(4) Adaptive P_c, P_m

In many research literatures, fixed P_c, P_m are used in the application of variable algorithm. When their values are too small, the population is affected by mutation operator and crossover operator very little, which leads to the difficulty of new gene integration; When the value is too high, the original excellent genes in the population may be destroyed. Therefore, the values of P_c and P_m should be dynamically balanced to increase the too small value and reduce the too large value. The following is the formula of P_c, P_m:

$$P_c = \begin{cases} P_c \cdot (F_{\max} - X_{\max})/(F_{\max} - F_{\mathrm{arg}}), & X_{\max} \geq F_{\mathrm{arg}} \\ P_c, \quad X_{\max} < F_{\mathrm{arg}} \end{cases} \tag{5}$$

$$P_m = \begin{cases} P_m \cdot (F_{\max} - Y_{\max})/(F_{\max} - F_{\mathrm{arg}}), & X_{\max} \geq F_{\mathrm{arg}} \\ P_m, \quad Y_{\max} < F_{\mathrm{arg}} \end{cases} \tag{6}$$

In the above formula, $X_{\max}$ represents the larger median of the two individuals, and Y is the fitness value calculated by the individual after mutation.

5 EXPERIMENTAL RESULTS & ANALYSIS

The data source we use is a school's performance database. We set the computer memory at 8GB and 16GB to simulate the actual effect. The model of the simulation platform is matlab2018b, and the capacity of the data set used in the experiment is 1GB, 2GB, 4GB, 8GB, 16GB, 32GB, 64GB. When the minimum support is 0.1 and the number of partition is 8, the Apriori-GA algorithm is compared with Apriori algorithm, and the results are shown in Figure 3. On the other hand, the simulation results of the impact of the number of partition in the data set on the time consumption are shown in Figure. 4.

As can be seen from Figure 3, if the computer memory (CM) of the running system is 8GB, the performance of Apriori-GA algorithm is close to that of Apriori algorithm. The former has a slightly higher mining efficiency than the latter, and the larger the data set, the closer the performance. When the CM is 16 GB, the larger the data set, the mining efficiency of Apriori-GA algorithm is much higher than that of Apriori algorithm.

As can be seen from Figure 4, when the data set is partitioned, the algorithm with less partition times consumes more time. In this experiment, after 8 partitions, the time consumption is far less than 4 partitions. Therefore, the number of times of data set partition has a great impact on the performance, and the time required by the algorithm does not show a positive correlation with the number of partitions. This shows that Apriori needs to access the database frequently when calculating, and the number of candidate sets generated is also huge. Therefore, the efficiency of Apriori algorithm is greatly reduced. Apriori-GA can find the global optimal solution in the set of association rules because it reduces the number of access to the database. As the data continues to grow, the performance of Apriori-GA algorithm gradually surpasses that of Apriori algorithm. Based on the above experimental results, Apriori-GA algorithm has a high value in large database data mining.

6 SUMMARY

In this paper, the shortcomings of Apriori algorithm are improved. Genetic algorithm and partition algorithm are introduced to establish association rule mining model, which improves the efficiency and accuracy of the algorithm. In the process of improvement, corresponding solutions are proposed for coding scheme and fitness function, and the practicability of Apriori-GA algorithm in big data mining is verified by simulation test.

REFERENCES

[1] Sanjay Rathee, Manohar Kaul, and Arti Kashyap. 2015. R-Apriori: An Efficient Apriori based Algorithm on Spark. In Proceedings of the 8th Workshop on Ph.D. Workshop in Information and Knowledge Management (PIKM '15). Association for Computing Machinery, New York, NY, USA, 27–34. DOI: https://doi.org/10.1145/2809890.2809893

[2] Kun Niu, Haizhen Jiao, Zhipeng Gao, Cheng Chen, and Huiyang Zhang. 2017. A developed apriori algorithm based on frequent matrix. In Proceedings of the 5th International Conference on Bioinformatics and Computational Biology (ICBCB '17). Association for Computing Machinery, New York, NY, USA, 55–58. DOI: https://doi.org/10.1145/3035012.3035019

[3] Mercy Mlambo, Naison Gasela, Michael Esiefarienrhe, and Bassey Isong. 2017. On the Optimization of Improved Apriori Algorithm via Linked-list Trie. In Proceedings of the 2017 International Conference on Big Data Research (ICBDR 2017). Association for Computing Machinery, New York, NY, USA, 62–66. DOI: https://doi.org/10.1145/3152723.3152740

[4] Aruna Govada, Abhinav Patluri, and Atmika Honnalgere. 2017. Association Rule Mining using Apriori for Large and Growing Datasets under Hadoop. In Proceedings of the 2017 VI International Conference on Network, Communication and Computing (ICNCC 2017). Association for Computing Machinery, New York, NY, USA, 14–17. DOI: https://doi.org/10.1145/3171592.3171621

[5] Iqra Jahangir, Abdul-Basit, Abdul Hannan, and Sameen Javed. 2018. Prediction of Dengue Disease through Data Mining by using Modified Apriori Algorithm. In Proceedings of the 4th ACM International Conference of Computing for Engineering and Sciences (ICCES'18). Association for Computing Machinery, New York, NY, USA, Article 5, 1–4. DOI: https://doi.org/10.1145/3213187.3287612

[6] Sherimon P. C. and Vinu Sherimon. 2017. A proposed onto-Apriori algorithm to mine frequent patterns of high quality seafood. In Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing (ICC '17). Association for Computing Machinery, New York, NY, USA, Article 169, 1–6. DOI: https://doi.org/10.1145/3018896.3056786

[7] Yuan Jinhui, Zhou Hongwei, and Zhang Laishun. 2019. Anomaly Event Detection for Sensor Networks on Apriori Algorithms and Subjective Logic. In Proceedings of the 2019 8th International Conference on Software and Computer Applications (ICSCA '19). Association for Computing Machinery, New York, NY, USA, 560–563. DOI: https://doi.org/10.1145/3316615.3316679

[8] Suzhen Wang and Haowei Zhou. 2016. The research of mapreduce load balancing based on multiple partition algorithm. In Proceedings of the 9th International Conference on Utility and Cloud Computing (UCC '16). Association for Computing Machinery, New York, NY, USA, 339–342. DOI: https://doi.org/10.1145/2996890.3007886

[9] David D. Linz, Hao Huang, and Zelda B. Zabinsky. 2016. A quantile-based nested partition algorithm for black-box functions on a continuous domain. In Proceedings of the 2016 Winter Simulation Conference (WSC '16). IEEE Press, 638–648.

[10] V. E. Torchinskii, O. S. Logunova, N. S. Sibileva, and P. Yu. Romanov. 2018. Genetic algorithm modification: addition of the population improvement stage. In Proceedings of the 2018 International Conference on Information Science and System (ICISS '18). Association for Computing Machinery, New York, NY, USA, 286–290. DOI: https://doi.org/10.1145/3209914.3209928

[11] Haipeng Li, Cuie Zheng, and Jucheng Zhang. 2018. Redundant Dictionary Construction via Genetic Algorithm. In Proceedings of the 2nd International Conference on Vision, Image and Signal Processing (ICVISP 2018). Association for Computing Machinery, New York, NY, USA, Article 66, 1–5. DOI:https://doi.org/10.1145/3271553.3271604

[12] Shaymaa Al-hayali, Osman Ucan, and Oguz Bayat. 2018. Genetic Algorithm for finding shortest paths Problem. In Proceedings of the Fourth International Conference on Engineering & MIS 2018 (ICEMIS '18). Association for Computing Machinery, New York, NY, USA, Article 27, 1–6. DOI: https://doi.org/10.1145/3234698.323472