Combinatorial Association Mining Method for High-Dimensional Data

jianwu Jiang
Guilin University of Technology
xu Gong
Guilin University of Technology
jingwen Li (lijw@glut.edu.cn)
Guilin University of Technology
yanling Lu
Guilin University of Technology

Research Article

Keywords: association rule mining, data mining, data analysis, data filtering, big data

Posted Date: October 15th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-944274/v1

COMBINATORIAL ASSOCIATION MINING METHOD FOR HIGH-DIMENSIONAL DATA

jianwu Jiang[1], xu Gong[1], jingwen Li [1,*], and yanling Lu[1]

1 Guilin University of Technology, College of Geomatics and Geoinformation, Guilin, 541004, China

* corresponding. lijw@glut.edu.cn

Abstract

Aiming at the problems of low mining efficiency, strong subjectivity and too many association relations generated by classical mining algorithms, a novel algorithm for association rule mining of high-dimensional data is designed in terms of both sample selection and association rule generation. The algorithm reduces the impact of weak samples at the beginning of mining by calculating the distribution coefficients and deletion thresholds of the samples and synthesizing custom support to double screen the samples at the first reading of the dataset. When generating frequent items and association rules, the algorithm mines information in a sample relationship table and sample the full relationship combination mode, which reduces the complexity and resource consumption of the mining process. The experimental results show that the number of frequent items and association rules mined by the Marc algorithm is significantly reduced, and the mining efficiency and memory consumption of the Marc algorithm are better than those of the Apriori, FP growth and Eclat algorithms. The higher the dimension is, the larger the data set is, and the more obvious the advantage is. The accuracy of the Marc algorithm for mining frequent items and association relations is 100%.

Keywords: association rule mining, data mining, data analysis, data filtering, big data

1 Introduction

Association rule mining[1] is one of the main tasks of data mining[2,3] and one of the most important and active research areas in the field of data mining[4]. It was first applied to customer behavior mining, and now it is gradually extended to various fields[5,6], especially in the analysis of users' business behavior, which has very important practical significance. The classic association rule mining algorithms mainly include Apriori[7,8] algorithm, FP-Growth[9,10] algorithm and Eclat[11,12] algorithm, the Apriori algorithm requires multiple scans of the dataset and suffers from low mining efficiency and redundancy of candidate sets[13,14] , although FP growth reduces the number of data scans and improves the mining efficiency[15,16] , it also has the problem of generating a large number of headers and repeatedly building FP Tree, which leads to serious memory consumption[13], Eclat algorithm is a vertical data format mining algorithm that uses depth-first mode for frequent pattern mining without repeated traversal of the data[17], but when the data scale is large to solve the support degree, it will consume many memory and reduce the mining efficiency[18]. In addition, the classic association rule mining algorithm[19] requires artificial setting of support and confidence, which is subjective and affects the effect of data mining[20-22]. One of the key factors to improve the efficiency of mining is to

effectively filter the data according to the distribution of the data set[23]. Second, in most cases, users are only interested in a certain part of the relationship between samples, but Apriori, FP growth and Eclat all try their best to mine all the association rules in the sample set[24], which reduces the accuracy of the mining algorithm. The quality of the association rule mining algorithm is closely related to the mining efficiency, memory consumption and algorithm complexity[25]. However, according to the experimental results of this paper, when the sample set dimension exceeds 15, the mining efficiency of the above three algorithms is significantly reduced, the memory consumption also increases sharply, and the number of association rules mined is very large. In response to the above problems, this paper proposes a Mining Multidimensional Association Rules By Combination (Marc) algorithm suitable for high-dimensional data. The core idea of the algorithm is to introduce the concept of overall data distribution, calculate the distribution coefficient and delete threshold, increase the data screening process, build a sample relationship table, and combine the minimum support and minimum confidence set by users to mine the data. After experimental analysis, the algorithm is better than the Apriori, FP-Growth and Eclat algorithms in mining efficiency, memory consumption, number of generated association rules and number of generated frequent items, and it is more in line with data expectations.

2 Principle of Marc algorithm

The Marc algorithm mainly solves the problem of high-dimensional data mining. When dealing with samples, it is mainly based on the influence of the distribution of samples on frequent items and association rules and the premise that the expansion of full permutation of samples does not change the actual frequent items and association rules. Based on the above premise, the innovation of the Marc algorithm has the following two aspects.

(1) Secondary filtering during data reading

The Apriori, FP-Growth and Eclat algorithms perform data filtering based on the minimum support set by the user during the initial read of the data, which is more subjective. The Marc algorithm introduces two indicators: the sample set distribution coefficient and deletion threshold. When the data are read for the first time, secondary screening is added to the initial samples according to the distribution of the samples. The occurrence frequency of the reserved samples must be greater than the minimum support and deletion threshold at the same time. This step is conducive to reducing the number of weak frequent items and weak association to improve the efficiency and focus of mining.

(2) Reconstruction of sample combinations based on a sample relation table

The core of the Marc algorithm mining frequent items and association relations is to construct the relationship table of samples. Then, according to the relationship table, the samples in the sample set are fully arranged and combined, the dimension of the sample set is expanded, the association relations between samples are transformed into full relationship combinations, and then the combined samples are mined. Compared with FP-Tree organization, the all-relational approach takes up less memory, has lower processing complexity, and is more convenient for calculating support and confidence in high-dimensional data mining, which is more conducive to the mining of massive high-dimensional data.

2.1 Algorithm index

(1) Support: Indicates the percentage of all transactions that contain both A and B.

$$Support(A \& B) = P(A \& B) \tag{1}$$

(2) Confidence: Indicates the proportion of transactions that use A that also contains B, i.e., the proportion of transactions that contain both A and B to the proportion of transactions that contain A.

$$Confidence(A \& B / A) = P(A \& B) / P(A) \tag{2}$$

(3) Distribution: The distribution coefficients are used to calibrate the dispersion of the samples. The distribution coefficients of the Marc algorithm are obtained using SigMod variance or standard deviation planning of the samples.

$$Distribution = \frac{1}{1+e^{-k}} \tag{3}$$

The value of k is the variance or standard deviation of the sample, and the specific choice is determined according to the actual needs.

(4) Trim Threshold: It is obtained by the product of the distribution coefficient Distribution and the sample mean E. It is mainly used to remove the weakly frequent terms in the sample.

$$TrimThreshold = Distribution \times E \tag{4}$$

2.2 Procedure

The core idea of the Marc algorithm is to consider the overall distribution of data when screening data, build a relationship table between samples, recombine the sample data based on the relationship table, and mine frequent items and association relationships. The mining process of the Marc algorithm is shown in Fig. 1.

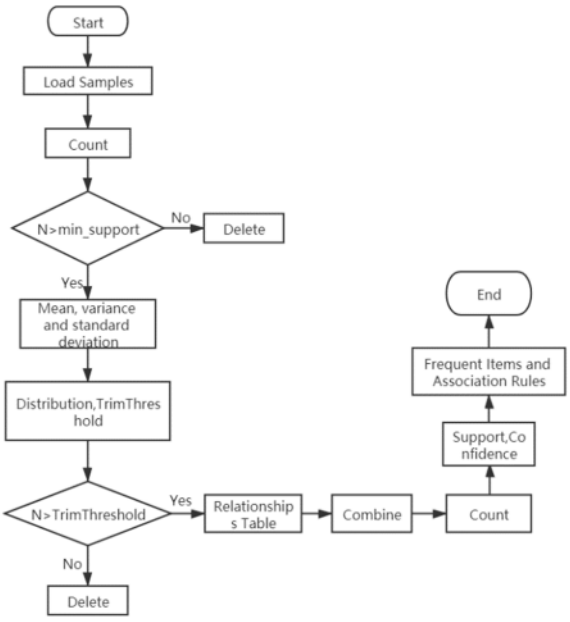


Fig. 1 Marc algorithm flow chart

As shown in Fig. 1, the Marc algorithm is mainly divided into five parts: sample secondary screening, sample frequency statistics and sorting, sample relation table generation, sample full

permutation and combination, and frequent item and association relationship generation.

(1) According to the minimum support and the calculated deletion threshold, the original data set is filtered, and the data set is arranged in descending order according to the sample frequency.

(2) Scan the sorted data set, record a single sample and the number of occurrences of a single sample to form a sample frequency set. Then, the samples in the sample frequency set are combined two by two, the number of occurrences of the sample combinations are counted separately, and the sample combinations with less than the minimum support are deleted to form the sample combination frequency set.

(3) Generation of sample relationship tables based on sample combination frequency sets.

(4) Scan the sorted data set generated in (1) and rearrange the samples according to the sample relation table to form a new sample data set.

(5) Scan the dataset generated in step (4), count the frequency of combined samples, calculate the support and confidence of combined samples, and generate frequent items and frequent patterns.

The detailed steps of the Marc algorithm are as follows:

(1) Set the minimum support to min_support and the minimum confidence to min_confidence. I represents the set of samples with m rows and p dimensions.

$$\mathrm{I} = [\mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3, \ldots, \mathrm{I}_\mathrm{m}]^\mathrm{T} \tag{5}$$

$$I_i = \left[x_{i1}, x_{i2}, x_{i3}, \ldots, x_{ij}\right] \qquad \{1 \le i \le m \ , 1 \le \ j \le p\} \tag{6}$$

where p is the largest dimension and I_{ij} contains samples less than or equal to p.

(2) Traversing the sample set I, count the number of each sample x_{ij} in the sample set, marked as N_{ij}. The total number of samples is M.

$$N_{ij} = \sum_{i=1}^{m} \sum_{j=1}^{p} Count\left(x_{ij}\right) \tag{7}$$

$$M = \sum_{i=1}^{m} \sum_{j=1}^{p} Count\left(N_{ij}\right) \tag{8}$$

As shown in Fig.2,Delete N_{ij} is less than the minimum support of the sample, and according to N_{ij} in descending order.

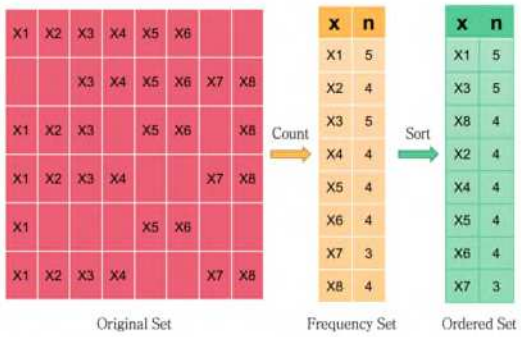


Fig. 2 sample frequency statistics and sorting

(3) Calculate the mean E, variance σ^2 and standard deviation S of the sample frequencies.

$$E = \frac{1}{M}\sum_{k=1}^{M} N_k \qquad \{k \in (i,j), 1 \le i \le m \ , 1 \le \ j \le p\} \tag{9}$$

$$\sigma^2 = \frac{\sum_{k=1}^{M}(N_k - E)^2}{M} \tag{10}$$

$$S = \sqrt{\sigma^2} \tag{11}$$

(4) Use the SigMod function to plot the standard deviation or variance to derive the distribution coefficient σ_d.

$$\sigma_d = \frac{1}{1+e^{-Index}} \qquad \{Index \in (S, \sigma^2)\} \tag{12}$$

Since the variance and standard deviation reflect the degree of sample dispersion, they can be used as indicators to screen the original sample. The purpose of planning by SigMod is to plan the dispersion of this sample between [0,1], which is more convenient when solving for the deletion threshold. The specific index used to calculate the distribution coefficient can be determined according to the actual situation. It should be noted that since SigMod has a small range of variables, as shown in Fig. 3, the results are very close to 1 when the variable value is less than -6 or greater than 6. Therefore, to better filter the samples, it is recommended to select the variance when the sample variance is large and the standard deviation when the sample variance is small.

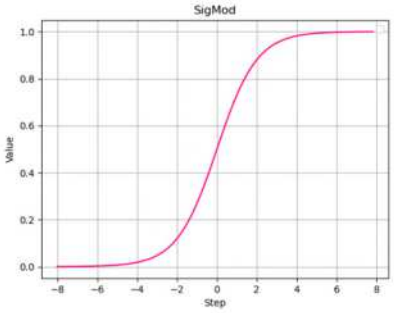


Fig. 3 SigMod function value change curve

(5) Calculate the deletion threshold σ_{trim}, delete all samples with N_{ij} less than σ_{trim} in the sample set, and obtain a new sample set I'.

$$\sigma_{trim} = E \times \sigma_d \tag{13}$$

(6) Traverse the sample set I', and calculate the number of occurrences of each sample x_{ij} in the sample $N_{ij}{}'$ to form the sample frequency set $Samples$. Let the total number of different samples x_{ij} be M', and since some sample data are deleted, the number of samples and dimensions at this point are less than or equal to the original number of samples and dimensions, the new number of samples is denoted as m' and the dimensions are denoted as p'.

$$N_{ij}{}' = \sum_{i=1}^{m'}\sum_{j=1}^{p'} Count(x_{ij}) \qquad \{1 \le i \le m', 1 \le j \le p'\} \tag{14}$$

$$M' = \sum_{i=1}^{m'}\sum_{j=1}^{p'} N_{ij} \tag{15}$$

$$Samples = [\{s_1 : c_1\}, \{s_2 : c_2\}, \ldots, \{s_i : c_i\}] \ \{1 \le i \le p'\} \tag{16}$$

s_i represents the i-th sample, and c_i represents the frequency of the i-th sample.

(7) The samples in the $Samples$ set are combined in pairs to form a new set, which is recorded as $Samples'$. Count the frequency of the combined samples in the set I', and delete the samples that are less than the minimum support.

$$Samples' = \left[\{s_{i1}: c_{i1}\}, \{s_{i2}: c_{i2}\}, \ldots, \{s_{ij}: c_{ij}\}\right]\{1 \le i \le p', 1 \le j \le p', i \ne j\} \quad (17)$$

s_{ij} represents the combination of the i-th sample and the j-th sample, and c_{ij} represents the frequency of the i-th sample and the j-th sample.

(8) According to the set of $Samples'$, generate the sample relationship table $Samples'$. The specific generation method is based on a sample in the combination of samples to find all the samples related to it that exists in the $Samples'$ set.

$$Relations = \{s_1: [s_2, s_3, \ldots, s_i], s_2: [s_1, s_3, \ldots, s_i], \ldots, s_i: [s_1, s_2, \ldots, s_{i-1}]\} \quad (18)$$

(9) According to the sample relationship table $Relations$, each group of samples in I' is fully permuted and combined. The specific method is based on a sample in I', traverse other samples in I', if the sample is in the relationship table of the basic sample, then combine, loop all the samples in I' to form a full permutation set I'', at this time, the dimension of each group of data in I' may expand dramatically. Let the number of samples at this time be m'' and the dimension be p''.

$$\mathrm{I}'' = [\mathrm{I_1}'', \mathrm{I_2}'', \mathrm{I_3}'', \ldots, \mathrm{I_m}'']^{\mathrm{T}} \quad (19)$$

$$I_i'' = \left[x_{i1}', x_{i2}', x_{i3}', \ldots, x_{ij}'\right] \qquad \{1 \le i \in m'' \ , 1 \le \ j \le p''\} \quad (20)$$

$$x_{ij}' = \left[x_{i1}, x_{i2}x_{i3}, \ldots, x_{ip'}\right] \quad (21)$$

x_{ij}' is composed of multiple x_{ij}, m'' is equal to m', and $p'' \ge p''$.

(10) Traverse the sample set I'', calculate the number of occurrences of each sample x_{ij}' in the sample N_{ij}'', and form a new sample frequency set $Samples'$.

$$N_{ij}'' = \sum_{i=1}^{m''} \sum_{j=1}^{p''} Count\left(x_{ij}'\right) \qquad \{1 \le i \le m'', 1 \le j \le p''\} \quad (22)$$

$$M' = \sum_{i=1}^{m'} \sum_{j=1}^{p''} N_{ij}'' \quad (23)$$

$$Samples' = [\{s_1': c_1'\}, \{s_2': c_2'\}, \ldots, \{s_i': c_i'\}] \ \{1 \le i \le p''\} \quad (24)$$

(11) According to $Samples'$ calculate the probability of each new sample x_{ij}' appearing in $Samples'$, it is recorded as P_{ij}, the support of different sample combinations is calculated as $Support_{ij}$, and the confidence is recorded as $Confidence_{ij}$, delete samples smaller than the minimum support and minimum confidence.

(12) Generate frequent item sets and association relationship sets, arranged in descending order of support and confidence.

The process of mining frequent rules by Marc algorithm is shown in Fig.4.

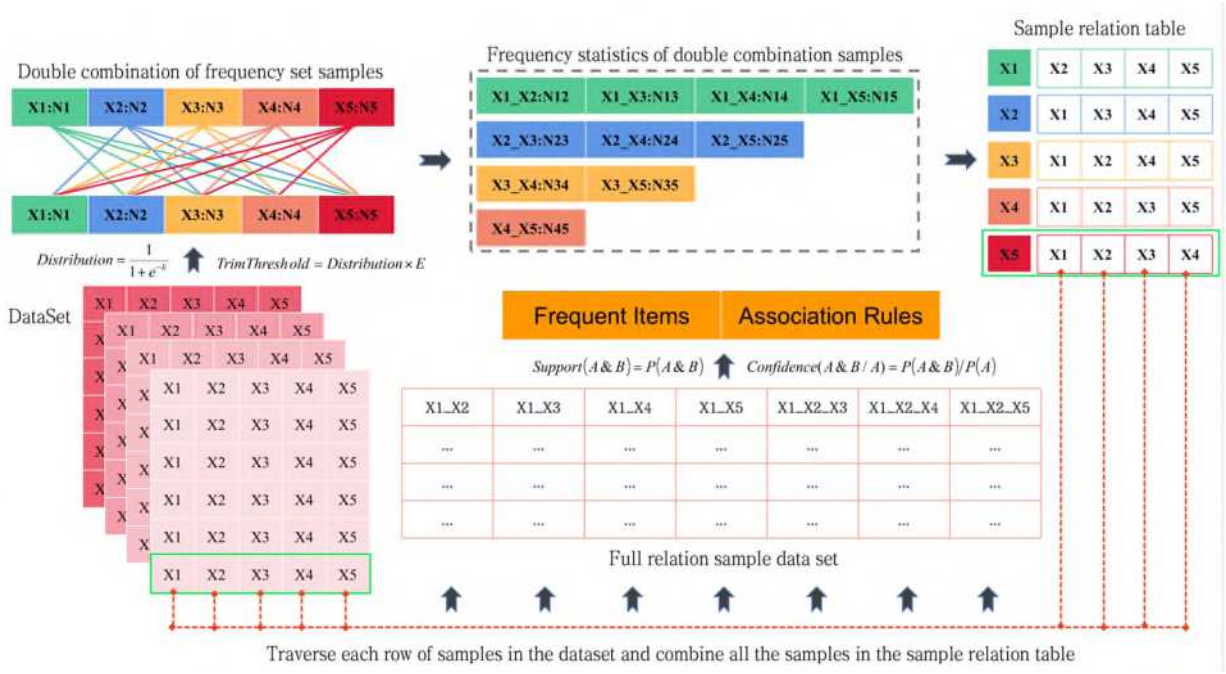


Fig. 4 Schematic diagram of the Marc algorithm mining frequent rules

3 Experimental analysis

To facilitate data mining, the degree of support is replaced by the degree of support. The minimum degree of support set in this experiment is 5, and the minimum confidence is 0.5. The Python version of the experimental environment is 3.8.5. The experimental equipment configuration is shown in Table 1.

Table 1 Experimental equipment configuration

Type	Parameters
CPU	Intel Core i7-7700K 4.2 GHz
Memory	16 GB 2400 MHz
OS	Microsoft Windows 10 (64 bit)

3.1 Experiment procedure

(1) Build a sample set

Construct 75 groups of test sample set, the maximum number of rows in each sample set is 300, the highest dimension of each row sample is 25, the number of rows starts from 20, and each increase of 20 rows is a group, and the dimension starts from 10 and each increase of 5 dimensions is one group. The specific generation method is as follows: first generate 300 rows of 10- to 25-dimensional samples, and then divide the samples according to the number of rows based on the samples.

To focus the experimental samples, their generation follows the following rules:

① Assuming that the odd-numbered items and odd-numbered items in the sample are subscripted, the even-numbered items are related to the even-numbered items, and the sample set is generated alternately according to odd and even;

② Set the maximum number of dimensions of the sample set to dimensions, and the total

number of samples in a row of records is at most dimensions;

③ The number of noise samples in each row is noise, and the noise value is randomly set between 0 and 3. The selection rule for noise samples is that when generating odd samples, the noise samples are even samples, and when generating even samples, the noise samples are odd samples.

④ The samples in the sample set of the same row are not repeated.

(2) Marc, Apriori, FP-Growth and Eclat were used to mine the above test data, and the mining time, memory consumption, number of frequent items and number of association rules generated were recorded.

(3) Analyze the experimental results and analyze the results of (2).

3.2 Analysis

This paper selects four indicators of algorithm execution time, memory consumption, number of frequent items and number of frequent relationships to make a comparative analysis of MARC, Apriori, FP growth and Eclat.

(1) Time

The comparison of mining time of the four algorithms in different dimensions is as follows:

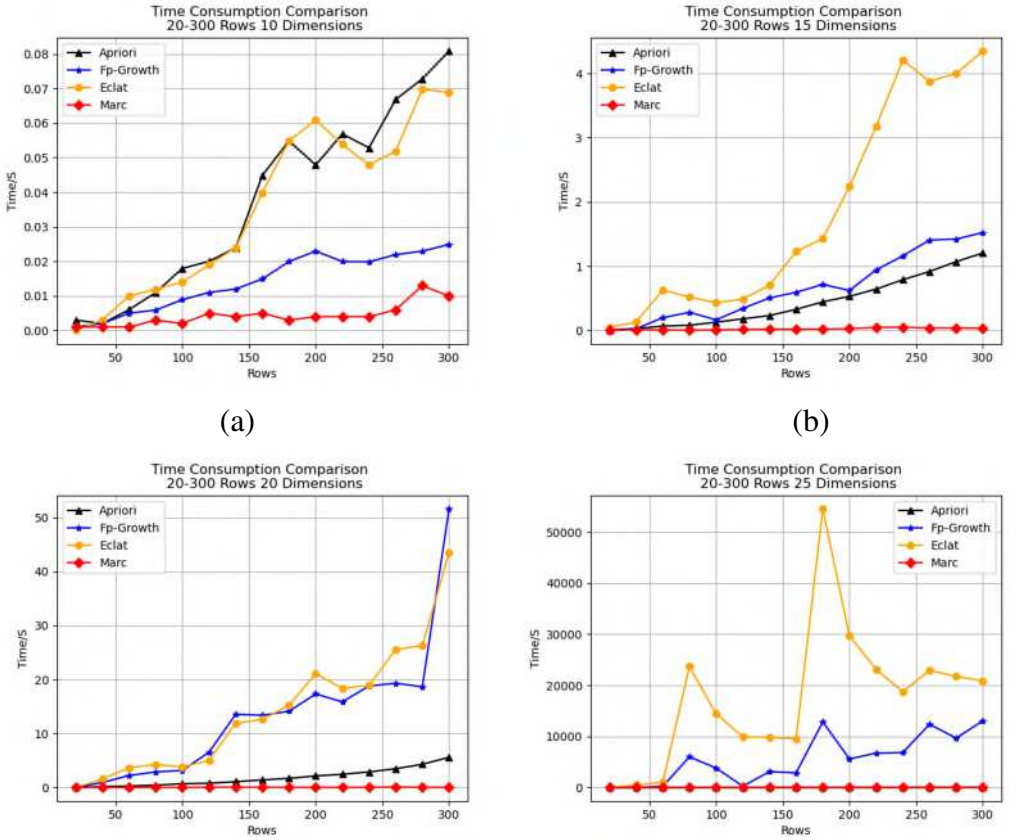


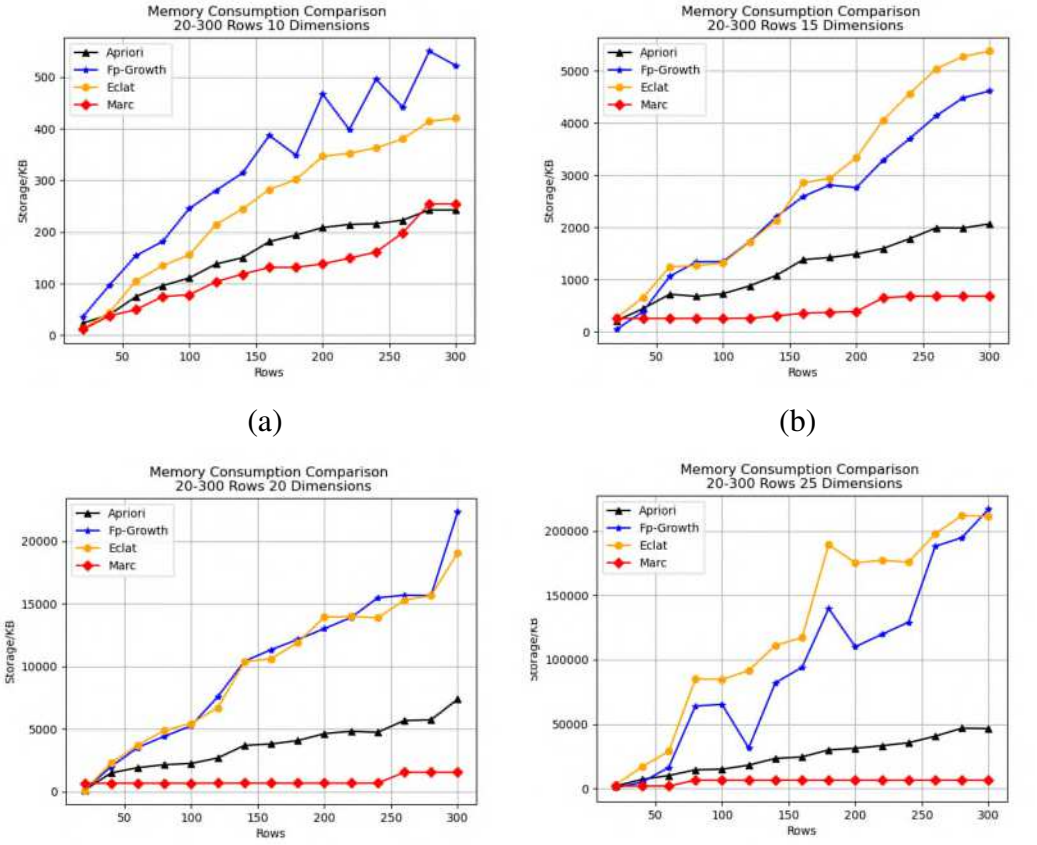
Fig. 5 Comparison of mining time

Although the dataset generation conditions are set, it still has a certain degree of randomness. Therefore, some test results show negative correlation characteristics. However, Fig. 5 shows that the mining time of the four algorithms is generally positively correlated with

the number of rows and dimensions. The mining time of the Marc algorithm is slightly higher than that of FP-Growth when the number of data rows is small and the dimensionality is low (as in Fig. 5a), which is because the Marc algorithm needs to generate sample relationship tables and sample full relationship combinations. Overall, the Marc algorithm has a lower running time than all the other three algorithms. In the other three algorithms, when the data dimension is 10, the time consumption of the Apriori algorithm is higher than that of FP growth and Eclat (as shown in Fig. 5a), but when the data dimension is increased, the execution time of the Apriori algorithm is lower than that of FP growth and Eclat. In terms of the growth rate of mining time, it can be seen from the figure that the growth rate of mining time of Marc algorithm and Apriori algorithm is more stable than that of FP growth and Eclat, while FP growth and Eclat significantly increase the mining time with the improvement of data dimension, and the mining time of peer number sample set increases exponentially with the growth of dimension. Through comparative analysis, it can be seen that compared with the other three algorithms, the Marc algorithm has higher mining efficiency, and the higher the data dimension and the number of rows, the more obvious the efficiency improvement. The FP-Growth and Eclat algorithms are suitable for mining low-dimensional data, the mining efficiency of both is lower than Apriori when the data dimension is higher than 10 dimensions, and the efficiency of Eclat mining is more volatile when the data dimension is 25.

(2) Memory

The comparison of memory consumption of the four algorithms is as follows:



(a) (b)

(c) (d)

Fig. 6 Comparison of memory consumption

As shown in Fig. 6, the memory consumed during Marc mining is generally smaller than that of the Apriori, FP-Growth and Eclat algorithms, and the higher the sample dimension is, the more obvious the memory advantage of the Marc algorithm. For low-dimensional data mining, Marc consumes more memory because it generates sample relational tables and full relational combinations, but for high-dimensional data mining, the other three algorithms' data organization formats consume much more memory than Marc's algorithm's relational tables and full relational combinations. In terms of the growth rate of memory consumption, the growth rate of the Marc algorithm is much smaller than that of the other three algorithms, and the growth rate of memory consumption of the Marc algorithm tends to be stable with increasing dimensionality and number of rows. Therefore, the resource consumption of the Marc algorithm in high-dimensional data mining is much lower than that of the other three algorithms, and the stability is higher than that of the other three algorithms.

(3) Frequent items

The comparison of the number of frequent items generated by the four algorithms is as follows:

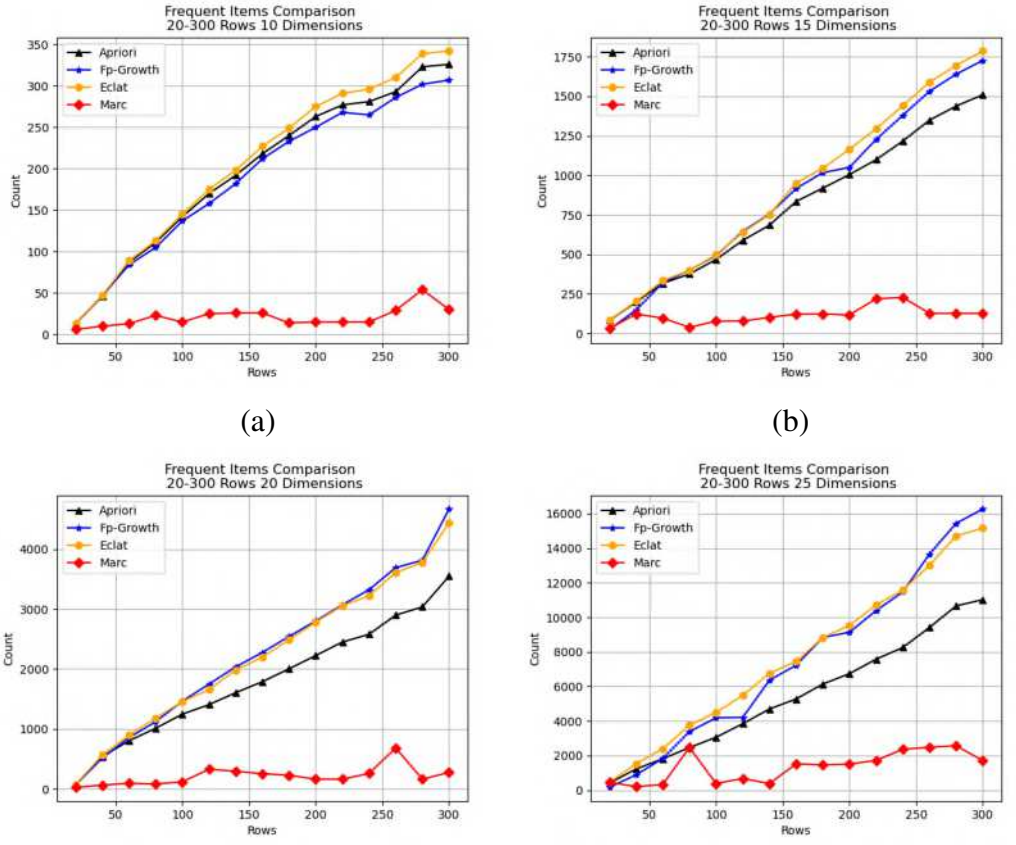


Fig. 7 Comparison of the number of frequent items

Fig. 7 shows that the number of frequent items mined by the four algorithms tends to stabilize as the number of sample rows increases. The number of frequent items mined by Apriori, FP-Growth and Eclat algorithms is comparable, and the number of frequent items mined by Marc algorithm is much lower than that of FP-Growth, Apriori and Eclat, and the number of frequent items mined by Marc algorithm gradually tends to be stable with the

increase of the number of rows at the same latitude, which is more in line with the actual situation of the data.

(4) Association rules

A comparison of the number of associations generated by the four algorithms is shown below:

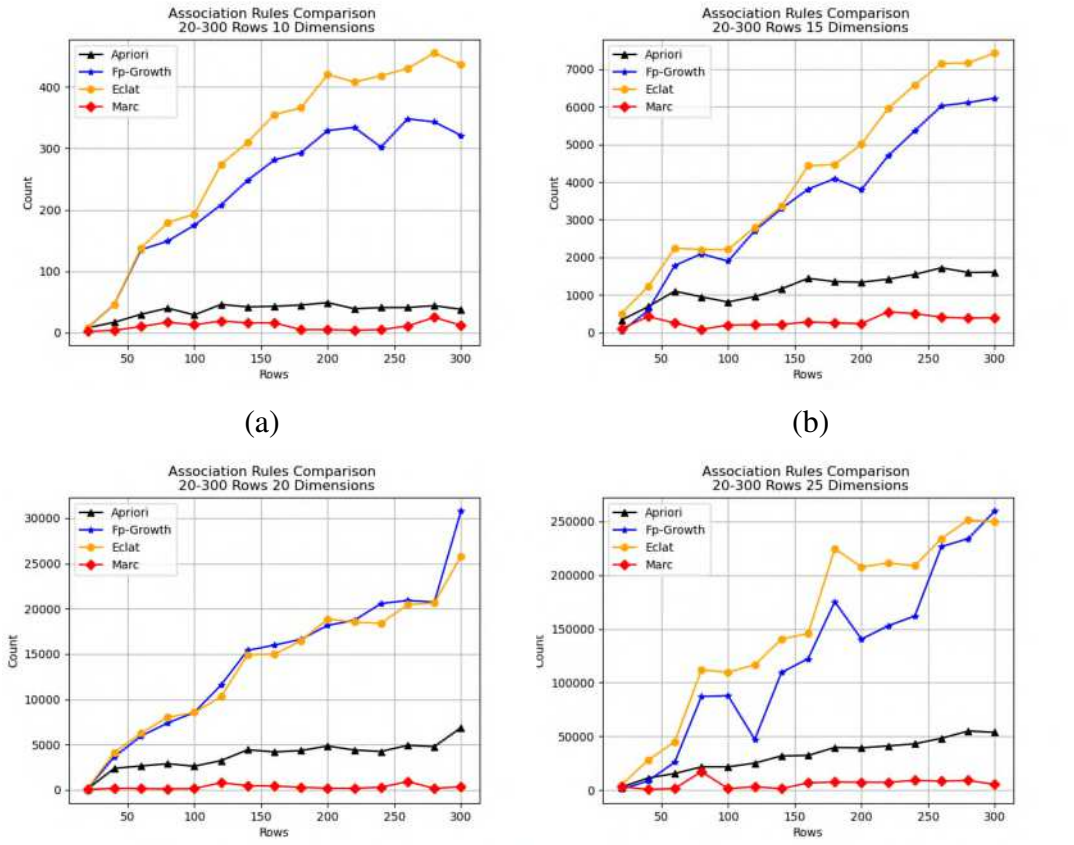


Fig. 8 Comparison of the number of associated rules

It can be seen from Fig. 8 that the number of association relationships mined by the Marc algorithm is less than that of the other three algorithms, and the higher the dimension is, the more obvious the gap. With the increase in the number of rows and dimensions, the number of association relationships generated by the other three algorithms also increases sharply. The association relationship generated by the Marc algorithm is relatively stable, and the higher the number of rows and the dimension, the higher the stability of the Marc algorithm. The Marc algorithm significantly reduces the number of association rules generated, and the more samples there are, the more stable the mined association relationship, which is more in line with the characteristics of the sample.

3.3 Accuracy

To facilitate the calculation, the verification sample set used to verify the mining accuracy is a 10-row 5-dimensional full-relational sample, and each row consists of five samples from X1 to X5. The minimum support degree set by the verification experiment is 10, and the minimum confidence degree is 0.5. Since it is a sample of full relationships, the support and

confidence of all relationships are 1. The actual frequent item of the verification sample is 31, and the actual number of association relationships is 129.

The frequent items and association rules mined by the Marc algorithm are as follows:

Table 2 List of frequent items

Frequent item	Frequent item	Frequent item	Frequent item	Frequent item	Frequent item
X1	X2	X3	X4	X5	X1_X2
X1_X3	X1_X4	X1_X5	X2_X3	X2_X4	X2_X5
X3_X4	X3_X5	X4_X5	X1_X2_X3	X1_X2_X4	X1_X2_X5
X1_X3_X4	X1_X3_X5	X1_X4_X5	X2_X3_X4	X2_X3_X5	X2_X4_X5
X3_X4_X5	X1_X2_X3_X4	X1_X2_X3_X5	X1_X2_X4_X5	X1_X3_X4_X5	X2_X3_X4_X5
X1_X2_X3_X4_X5					

Table 3 List of Association Rules

Association Rule	Association Rule	Association Rule	Association Rule
X1_X2&X1	X1_X2&X2	X1_X3&X1	X1_X3&X3
X1_X4&X1	X1_X4&X4	X1_X5&X1	X1_X5&X5
X2_X3&X2	X2_X3&X3	X2_X4&X2	X2_X4&X4
X2_X5&X2	X2_X5&X5	X3_X4&X3	X3_X4&X4
X3_X5&X3	X3_X5&X5	X4_X5&X4	X4_X5&X5
X1_X2_X3&X1	X1_X2_X3&X1_X2	X1_X2_X3&X2	X1_X2_X3&X2_X3
X1_X2_X3&X3	X1_X2_X4&X1	X1_X2_X4&X1_X2	X1_X2_X4&X2
X1_X2_X4&X2_X4	X1_X2_X4&X4	X1_X2_X5&X1	X1_X2_X5&X1_X2
X1_X2_X5&X2	X1_X2_X5&X2_X5	X1_X2_X5&X5	X1_X3_X4&X1
X1_X3_X4&X1_X3	X1_X3_X4&X3	X1_X3_X4&X3_X4	X1_X3_X4&X4
X1_X3_X5&X1	X1_X3_X5&X1_X3	X1_X3_X5&X3	X1_X3_X5&X3_X5
X1_X3_X5&X5	X1_X4_X5&X1	X1_X4_X5&X1_X4	X1_X4_X5&X4
X1_X4_X5&X4_X5	X1_X4_X5&X5	X2_X3_X4&X2	X2_X3_X4&X2_X3
X2_X3_X4&X3	X2_X3_X4&X3_X4	X2_X3_X4&X4	X2_X3_X5&X2
X2_X3_X5&X2_X3	X2_X3_X5&X3	X2_X3_X5&X3_X5	X2_X3_X5&X5
X2_X4_X5&X2	X2_X4_X5&X2_X4	X2_X4_X5&X4	X2_X4_X5&X4_X5
X2_X4_X5&X5	X3_X4_X5&X3	X3_X4_X5&X3_X4	X3_X4_X5&X4
X3_X4_X5&X4_X5	X3_X4_X5&X5	X1_X2_X3_X4&X1	X1_X2_X3_X4&X1_X2
X1_X2_X3_X4&X1_X2_X3	X1_X2_X3_X4&X2	X1_X2_X3_X4&X2_X3	X1_X2_X3_X4&X2_X3_X4
X1_X2_X3_X4&X3	X1_X2_X3_X4&X3_X4	X1_X2_X3_X4&X4	X1_X2_X3_X5&X1
X1_X2_X3_X5&X1_X2	X1_X2_X3_X5&X1_X2_X3	X1_X2_X3_X5&X2	X1_X2_X3_X5&X2_X3
X1_X2_X3_X5&X2_X3_X5	X1_X2_X3_X5&X3	X1_X2_X3_X5&X3_X5	X1_X2_X3_X5&X5
X1_X2_X4_X5&X1	X1_X2_X4_X5&X1_X2	X1_X2_X4_X5&X1_X2_X4	X1_X2_X4_X5&X2
X1_X2_X4_X5&X2_X4	X1_X2_X4_X5&X2_X4_X5	X1_X2_X4_X5&X4	X1_X2_X4_X5&X4_X5
X1_X2_X4_X5&X5	X1_X3_X4_X5&X1	X1_X3_X4_X5&X1_X3	X1_X3_X4_X5&X1_X3_X4
X1_X3_X4_X5&X3	X1_X3_X4_X5&X3_X4	X1_X3_X4_X5&X3_X4_X5	X1_X3_X4_X5&X4

X1_X3_X4_X5&X4_X5	X1_X3_X4_X5&X5	X2_X3_X4_X5&X2	X2_X3_X4_X5&X2_X3
X2_X3_X4_X5&X2_X3_X4	X2_X3_X4_X5&X3	X2_X3_X4_X5&X3_X4	X2_X3_X4_X5&X3_X4_X5
X2_X3_X4_X5&X4	X2_X3_X4_X5&X4_X5	X2_X3_X4_X5&X5	X1_X2_X3_X4_X5&X1
X1_X2_X3_X4_X5&X1_X2	X1_X2_X3_X4_X5&X1_X2_X3	X1_X2_X3_X4_X5&X1_X2_X3_X4	X1_X2_X3_X4_X5&X2
X1_X2_X3_X4_X5&X2_X3	X1_X2_X3_X4_X5&X2_X3_X4	X1_X2_X3_X4_X5&X2_X3_X4_X5	X1_X2_X3_X4_X5&X3
X1_X2_X3_X4_X5&X3_X4	X1_X2_X3_X4_X5&X3_X4_X5	X1_X2_X3_X4_X5&X4	X1_X2_X3_X4_X5&X4_X5
X1_X2_X3_X4_X5&X5			

From Table 2 and Table 3, it can be concluded that the accuracy of the Marc algorithm for mining the number of frequent items is 100%, and the accuracy of the mined association relationship is 100%.

4 Conclusion

The Marc Association Rule Mining algorithm is mainly for high-dimensional data mining scenarios. After experimental validation and comparative analysis, the following conclusions are drawn:

(1) Marc simplifies the time and space complexity of the mining task at the beginning of the mining process by introducing the distribution factor and deletion threshold index when the data are first filtered, greatly improving the mining efficiency of the mining algorithm and saving memory space.

(2) Marc builds the whole relationship combination of samples based on the sample relationship table, which simplifies the complexity of maintenance in the mining process and improves the efficiency of mining frequent items and related relationships.

(3) The Marc algorithm is better than the Apriori, FP-Growth and Eclat algorithms in terms of mining efficiency and memory consumption, and the generated frequent items and correlations are closer to the sample and application reality. The higher the latitude is, the more obvious the advantage, which is more suitable for massive and high-dimensional data mining.

(4) The accuracy of frequent items and association rule relationships mined by Marc's algorithm is 100%.

(5) The mining efficiency of the FP-Growth and Eclat algorithms is lower than that of Apriori when mining high-dimensional data.

References

1.Agrawal,R.,Imieliński,T.&Swami,A.N. Mining Association Rules between Sets of Items in Large Databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 22, 207-216 (1993).

2.Li,J.,Ma,X.,Zhang,J.,Tao,J.,Wang,P.&Guan X. Mining repeating pattern in packet arrivals: metrics, models, and applications. Information Sciences An International Journal, 408, 1-22(2017).

3.TeliKaNi,A.,Gandomi,A.H.&Shahbahrami,A. A survey of evolutionary computation for association rule mining. Information Sciences, 524 (2020).

4.GB,Baró.,JF,Martínez-Trinidad.,Rosas,R.,Ochoa,J.&MSL,Cortés. A pso-based algorithm for mining association rules using a guided exploration strategy. Pattern Recognition Letters, 138, 8-15(2020).

5.Li,T.,Li,Y.,An,D.,Han,Y.,Xu,S.,Lu,Z.&Crittenden,J. Mining of the association rules between

industrialization level and air quality to inform high-quality development in China. Journal of environmental management, 246, 564 – 574(2019).
6.He,H.,Yin,T.,Dong,J.,Zhang,P.&Ren,J. Efficient mining of high utility software behavior patterns from software executing traces. International journal of innovative computing, information & control: IJICIC, 11(5), 1779-1793(2015).
7.Agrawal,R. Mining association rules between sets of items in large databases. Acm Sigmod Conference on Management of Data (1993).
8.Agrawal,R.&Srikant,R. Fast Algorithm for Mining Association Rules in Large Databases. Proc. 20th Very Large Data Bases Conference. IEEE(1994).
9.Han,J.,Pei,J.,Yin,Y.&Mao,R. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Mining & Knowledge Discovery, 8(1), 53-87(2004).
10.Han,J.,&Pei,J. Mining frequent patterns without candidate generation. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA. ACM(2000).
11.Leung,C.K.S.&Sun,L. Equivalence class transformation based mining of frequent itemsets from uncertain data. ACM(2011).
12.Zaki,M.J. Scalable algorithms for association mining. IEEE Transactions on Knowledge & Data Engineering(2000).
13.Ruan,Zhi-Yi. Multiscale FP-Growth algorithm and its application on regular route mining. Acta Automatica Sinica[J].https://doi.org/10.16383/j.aas.c180316(2021).
14.Gu,Liang-Yun. Research on Frequent Pattern Mining Method[D]. Jiangnan University(2020).
15.Grahne,G.,&Zhu,J. Fast algorithms for frequent itemset mining using fp-trees. IEEE Transactions on Knowledge & Data Engineering(2005).
16.Ahmed,C.F.,Tanbeer,S.K.,Jeong,B.S.&Lee,Y.K. Efficient tree structures for high utility pattern mining in incremental databases. IEEE Transactions on Knowledge and Data Engineering, 21(12), 1708-1721(2009).
17.Yu,X.&Wang,H. Improvement of eclat algorithm based on support in frequent itemset mining. Journal of Computers, 9(9)(2014).
18.Feng,Xing-Jie.&Pan,Xuan. Eclat algorithm based on spark. Application Research of Computers(2019).
19.Song,A.,Ding,X.,Li,M.,Cao,W.&Pu,K. A novel binary bat algorithm for association rules mining. ICIC Express Letters, 9(9), 2387-2394(2015).
20.Yoo,J.S.&Bow,M. Mining spatial colocation patterns: a different framework. Data Mining & Knowledge Discovery, 24(1), 159-194(2012).
21.Qian,F.,He,Q.,Chiew,K.&He,J. Spatial colocation pattern discovery without thresholds. Knowledge and Information Systems, 33(2), 419-445(2012).
22.HE,Zhanjun.,DENG,Min.,CAI,Jiannan.&LIU,Qiliang. A context-based association rules mining method for multiple event sequences. Geomatics and Information ence of Wuhan University(2018).
23.Zaki,F.A.M.&Zulkurnain,N.F. Rare: mining colossal closed itemset in high dimensional data. Knowledge-Based Systems, 161(DEC.1), 1-11(2018).
24.Ls.A.,Fv,C.,Rya.B.&ld,A. A guided fp-growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data. Information Sciences(2020).
25.Xin,Chunhua.,Guo,Yanguang.&Lu,Xiaobo. Association rule mining algorithm using improving treap with interpolation algorithm in large database[J]. Application Research of Computers(01),88-92. doi:10.19734/j.issn.1001-3695.2019.11.0613(2021).

Acknowledgements

This study was supported by the National Natural Science Foundation of China (No. 41961063).

Author contributions statement

J.W.J is responsible for the innovation design, programming, thesis writing and modification of this manuscript.

X.G and Y.L.L are responsible for experimental verification.

J.W.L is responsible for revising the overall framework and opinion guidance.

Additional information

Marc algorithm and experimental data can be obtained through Gitee, URL: https://gitee.com/idmxh/marc.git

Apriori, FP-Growth and Eclat algorithm implementations are referenced from the GitHub open source project Machine Learning, URL: https://github.com/Ryuk17/MachineLearning/blob/master/AssociationAnalysis.py