Applying Correlation Threshold on Apriori Algorithm

Anand H.S.
Department of Computer Science
College of Engineering
Trivandrum, India

Vinodchandra S.S.
Computer Center
University of Kerala
Trivandrum, India

***Abstract*—Ever growing size of information and database has always demanded the scientific world for very efficient rule mining algorithm. This paper gives an extension to the Apriori algorithm, a classical rule mining algorithm. Apriori finds its application in areas of data mining, finding association between attributes and in prediction systems. Even though Apriori suits in various applications it possesses various disadvantages. To increase the efficiency of the present Apriori algorithm a method for incorporating a new correlation factor (threshold) is being introduced. First part of the paper provides a quick summary of basic Apriori algorithm and second half details the implementation of correlation threshold. Performance of the redesigned algorithm is evaluated and is compared with the traditional Apriori algorithm. The evaluation shows a peak improvement in the mining result. We reduce the time complexity of the newly designed algorithm into O (n). In an application level, qualitative content analysis of water was also conducted to affirm the results.**

Keywords—apriori algorithm; datamining; itemsets; correlation threshold; association rule mining; machine learning algorithm

I. INTRODUCTION

Data growth of the internet is increasing day to day. It is calculated that the indexed web contents counts to 9.77 billion pages and is still growing (the measure is just about the indexed web contents). This information provides an idea on the amount of data available in the web. Same is in the case of database, which is used for various applications. So, some efficient methods are needed to mine the data from large databases. This is one of the major difficulties of researchers working in the field of data mining. Every database has numerous attributes. Change in any of the attribute will affect other variables, which are closely associated with it. So it has always been an area of interest to know such interesting relationship between the various attributes within a database. Association rule mining is such a process which provides numerous ways to find association between variables. Consider a large database say, D having N attributes. Let A and B are any two variables, which are closely associated. Any variation in the value of A can cause a positive or negative effect on the associated variable B. These associations could be used to create or predict some rules or decisions. Hence it is an important area of study, to know the extent of association between such attributes. This is why association rule learning (ARL) is crucial. There are various algorithms which fall under ARL. Major association rule mining algorithms include Apriori algorithm, Tertius algorithm, Frequent pattern growth algorithm and Eclat algorithm. All these algorithms provide ways to create rules on associated attributes. This paper discusses the classical rule mining algorithm, Apriori [1, 2]. This algorithm suggests solutions to market basket analysis for finding the related products from a store. Such relationships can increase business and can be used for finding innovative methods for the advertisement of products. For example, consider the activities of Amazon online shopping store, when we browse a book under a particular stream, the related products will also populate in the side menu. It is merely by the concept of association mining. In every area, such association rules and data mining influence the business either directly or indirectly. We discuss methods for finding sharp associations with improved accuracy by incorporating correlation threshold to the existing algorithm.

II. CONCEPTS AND TERMINOLOGY

Two major concepts used while working with the Apriori algorithm is Support and Confidence [3]. Let's define what exactly these terms are:

Support: It defines the transactions where the item goes in hand by hand. If **a**, **b** are two itemsets, then the support can be defined as the transaction T which shows **a → b**

Confidence: It is defined as the percentage of transactions where the itemsets are most probable to occur. If **a**, **b** are two itemsets, then, the probability **a U b** is a subset of transaction, **T** is termed as the confidence.

In addition to the usual concepts we introduce a new term Correlation threshold, which is implemented in the proposed Apriori algorithm. Correlation threshold is a factor which transfers the probability from single itemset to n-itemset. This transition probability is required in-order to confirm the propagation of probability to each itemsets. The general equation for finding the correlation threshold ***c'*** is given by,

$$C' = [\alpha (P_{min} + \beta (P_{min} * P_{max})] \quad (1)$$

where P_{min} and P_{max} are the minimum and maximum probability of itemsets, which is calculated from the probabilistic array. In the formula we have two constants α and β; these are correlation constants whose value depends on

978-1-4673-5036-5/13/$31.00 © 2013 IEEE

the mean of probabilities. We need to make sure that $\alpha + \beta = 1$ in all situations.

Apriori property: The superset of all frequent itemset will be frequent [4, 5]. This is the major property used while calculating the frequent data.

Above mentioned properties make Apriori unique and classical from other association rule learning algorithms. By integrating support threshold, performance of the algorithm also increases.

III. THE APRIORI ALGORITHM

Basic Apriori algorithm is viewed as a two stage process. First, the candidate item set generation and then the rule creation. Before starting the procedure, the threshold **E** is defined. Algorithm starts by scanning the database, **D**. From **D** all the frequent items are obtained. First scan considers only single itemsets, successive iterations deals with 2-itemset. Thus new list of frequent items are created. The process continues till all the frequent itemsets are mined from **D**. Only those frequent items whose threshold is greater than **E** is taken for rule creation. The traditional Apriori function [6] is given in Algorithm 1.

Algorithm 1 Function Apriori

Join Step: C_k is generated by joining L_{k-1} with itself
Prune Step: Any (k-1) - item set that is not frequent cannot be a subset of a frequent k - item set.

```
Ck: Candidate item set of size k
Lk: frequent item set of size k
L1 = frequent items
for (k = 1; k < L; k++) do begin
  Ck+1 = candidates generated from Lk
  for each transaction t in database do
    increment the count of all candidates in
      Ck+1 that are contained in t
      Lk+1= candidates in Ck+1 with min support
    end
  return Ek U Lk
```

A. Limitationof Apriori Algorithm

While solving applications Apriori algorithm encounters with various drawbacks. For a given database **D**, Apriori needs to scan **n** times if the length of the frequent itemsets is **n** [7]. This extensive scan makes the system to consume more time for rule generation. Time complexity of such a process having **n** transactions with **m** itemset is defined as $O(e^n)$ [7]. Due to high threshold benchmark some of the frequent datasets are discarded in a very early step. This prunes significant predictions and rules. In cases where infrequent items are considered for rule generation, a weak prediction system is being created. To overcome all these short comings, we need a modification in the present Apriori algorithm. Major advent of this modification is the introduction of a correlation threshold.

IV. CORRELATION THRESHOLD

Correlation threshold finds its application in candidate item set generation. In modified Apriori, we incorporate correlation threshold for finding strong association rules between the itemsets. The correlation threshold is a value between 0 and 1. If the value is 1, then the attributes are highly related to each other. While a value close to zero shows the dataset as independent. This correlation confirms the presence of all itemset appearing in traditional Apriori in proposed algorithm. Algorithm begins by scanning the database, **D**. Suppose there are **n** elements in **D**, algorithm initializes a probabilistic array, **PA[n]**. After the first scan, probability of occurrence of each 1 itemsets appearing in the transaction is entered into **PA[n]**. From the probabilistic array, correlation threshold is found out using the equation (1). This acts as the minimum support threshold. Those itemsets whose threshold is below the correlation value is eliminated. This step repeats iteratively and 2-itemsets are generated from **PA[n]** by calculating new correlation threshold. Repeated database scan is being avoided as the candidate itemset generation is done directly from the database and not by continuous scanning of **D**. The process is continued till all attributes in the database is scanned. The pseudo code for the modified Apriori is given in Algorithm 2.

Algorithm 2 Modified Apriori

Input: Database **D** with **n** elements
Output: Frequent item, **I** and Rule **S**
Data structure Used: Probabilistic Array, $\mathbf{PA_0[n]}$

```
while (n!=NULL)
  Scan the database D
  Input the probability of the n elements into PA0[n]
  for (i=0;i<n; i++)
    Calculate correlation threshold Ci from PA0[n]
    for (k=0;k<n; k++)
      if (Ci'=PAi[k])
      update PAi+1[k]
   for each frequent item set I of non empty array PAi[n]
  if (Support(I) / Support(PAi[n] > c'i )
  generate rule s → (I- PAi[n])
```

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

Algorithm was used to predict the common minerals and other contents found in water. Table I denotes the real time database used for the content prediction. Drinking water contains various minerals and dissolved solids. By using this algorithm the most commonly occurring contents are found out. Table II is the 4-transaction database used to affirm the comparison results [9].

TABLE I. DATABASE – WATER CONTENTS

Transaction	*Dissolved Solids*	*CaC03*	*Cl*	*Nitrate*	*SO4*
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	0	1	0	0
8	1	1	1	0	1
9	1	1	1	0	0

TABLE II. 4- TRANSACTION DATABASE

Transaction	*Itemset 1*	*Itemset 2*	*Itemset 3*	*Itemset 4*	*Itemset 5*
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0

Stepwise process of the algorithm is explained by considering the database of water content (Table I). The attributes under consideration includes, dissolved solids, carbonates, chlorides, nitrates and sulphates. Each transaction is scanned one after the another and the probabilistic array **$PA_0[n]$** is populated. Now C_0' is calculated by equation (1). C_0' is the initial correlation threshold and is initialized to 2/9 which is obtained from the initial database. Correlation constant α is initialized to 5/9 which is the mean value of the probability of frequent itemsets. The data pruning is carried out by checking the **$PA_0[0]$** with C_0'. Again frequent itemsets are listed and the probabilistic array is modified by calculating a new correlation threshold. In second step, we need to find the 2-itemset threshold. From equation (1), the new C_1' is found out and the value was nearly, 0.20027. There are itemsets whose threshold is below C_1', they get pruned away. The process repeats until all the frequent items are visited. The correlation threshold in next step was obtained to be, 0.18049. Table III shows the various item sets which were obtained in addition to the candidate items generated by the Apriori algorithm. It clearly shows more itemsets were generated in the proposed algorithm in least time.

TABLE III. FREQUENT ITEM - ANALYSIS

Itemset	*Additional Candidate Item*
1- Itemset	No change
2–Itemset	Cl, $S0_4$
3- Itemset	$S0_4$, Cl, $CaCO_3$

A. Observation of 4-transaction database

The 4-Transaction database mentioned in Table II is a sample database commonly used for explaining Apriori algorithm. Table IV draws clear distinction on the number of itemsets generated by the traditional Apriori to the proposed algorithm. It is noted that the proposed algorithm generates more candidate keys. Let **a** be the difference in the number of itemsets generated. Even a small increase in the number of itemset can generate 2n (a-1) more rules [10]. Thus increases in the number of itemset will eventually step-up the number of rules created.

TABLE IV. FREQUENT ITEM SAMPLE DATABASE

Itemset	*Apriori Algorithm*	*Proposed Algorithm*
1 Itemset	1;2;3;5	1;2;3;4;5
2 Itemset	1,3 ; 2,3 ; 2,5 ; 3,5	1,2;1,3;1,4;1,5;2,3;2,5; 3,4;3,5;4,5
3 Itemset	2,3,5	2,3,5 ; 1,3,4

The comparison on the number of itemset generated is shown in Figure 1.

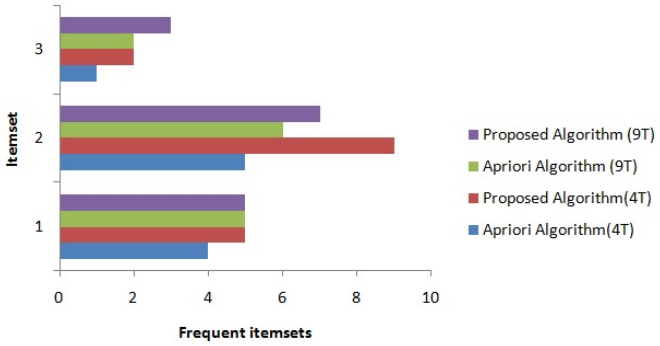

Fig. 1. COMPARISION OF ITEMSET

B. Time complexity of the algorithm

Time complexity of Apriori algorithm is the sum of time taken for generation of frequent items and time taken for rule generation. For n transactions with m itemsets, the time taken for frequent itemset generation is,

$$= n\ {}^{m}C_1 + n\ {}^{m}C_2 + \ldots..\ n\ {}^{m}C_m$$
$$= n \sum_{i=1}^{m} ({}^{m}C_i)$$
$$\approx O(e^n) \text{ as } n \to \infty \quad (2)$$

Similarly for rule generation the time complexity calculated as,

$$= {}^{k}C_1 + {}^{k}C_2 + \ldots.... + {}^{k}C_n$$
$$= \sum_{k=1}^{k=m} {}_{j=mCk}[\sum_{i=1}^{j} ({}^{j}C_i)$$
$$= \sum_{k=1}^{m} m \sum_{i=1}^{j} ({}^{j}C_i)$$
$$\approx O(n) \text{ as } n \to \infty \quad (3)$$

Thus total time complexity is the sum of (2) and (3), which shows the Apriori algorithm grows exponential,

$$\text{ie., } O(e^n) \text{ as } n \to \infty \quad (4)$$

For the proposed algorithm, the time complexity for frequent item generation is,

$$= \sum_{i=1}^{m}[(n-(m+1)) + (\sum_{0}^{m-1}(n-1)) + 1] \quad (5)$$

As the time complexity for rule generation is in O (n), the total time complexity is calculated as,

$$= \sum_{i=1}^{m}[(n-m) + \sum_{0}^{m-1}(n-1) + 2]$$

$$\text{ie., } O(n) \text{ as } n \to \infty \quad (6)$$

From this it is evident that the time complexity has been reduced to a great extent which increases the performance of the proposed algorithm. Throughout the experiment the confidence percentage was fixed to be 70. To have more accurate result, fix a confidence rate to the higher level. Higher the confidence rate, greater is the performance of the algorithm.

VI. CONCLUSIONS

The proposed algorithm suggests a correlation approach to the traditional Apriori. Method enhances the efficiency of algorithm by evaluating the number of candidate itemsets generated. Proposed scheme creates more number of frequent itemsets in lesser time. The time complexity was found to reduce from O(en) to O(n). Time complexity was reduced due to the abatement of the number of database scan. Through pruning the infrequent itemsets and by retaining the frequent ones strong rules are created. Database scan which was fully depended on the length of frequent itemset was supplanted by the introduction of probabilistic array. Results confirm that, with extended inter-transactional association, absolute and remarkable relations were able to mine

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," In Proc. of VLDB, pp. 487–499, 1994.

[2] L. Zeng, Q.He, and Z. Shi, "Parallel implementation of apriori algorithm based on mapreduce," In Proc. Of SNPD, pp.236-241, Aug 2012.

[3] H. Wu, Z. Lu, L. Pan, R. Xu, and W. Jiang, "An improved apriori-based algorithm for association rules mining," In Proc. of sixth international conference on fuzzy systems and knowledge discovery, pp. 51–55, 2009.

[4] S. Tao and P. Gupta, "Implementing improved algorithm over apriori data mining association rule algorithm," IJCST, vol. 3, pp. 489 – 493, Jan-Mar 2012.

[5] C. Cooper and M. Zito, "Realistic synthetic data for testing association rule mining algorithms for market basket databases," Knowledge Discovery in Databases: PKDD, vol. 9, pp. 398– 405, July-August 2007.

[6] L. Shi, J. niu Bai, and Y. lin Zhao, "Mining association rules based on apriori algorithm and application," In Proc. of IFCSTA, vol. 3, pp. 141–145, Dec 2009.

[7] A. S. Varde, M. Takahashi, E. A. Rundensteiner, M. O. Ward,M. Maniruzzaman, and R. D. Sisson, "Apriori algorithm and game of life for predictive analysis in materials science," International Journal of Knowledge based and Intelligent Engineering Systems, vol. 8, pp. 1 –16, 2004.

[8] D. Sun, S. Teng, W. Zhang, and H. Zhu, "An algorithm to improve the e_ectiveness of aprior," In Proc. of 6th IEEE International Conference on Cognitive Informatics, vol. 1, pp. 385–390, Aug 2007.

[9] K. Suneetha and R. Krishnamoorti, "Advanced version of apriori algorithm," In Proc. of ICIIC, vol. 5, pp. 238–245, Aug 2010.

[10] S. Nandagopal, "Mining of meteorolofical data using modified apriori algorithm," European Journal of Scientific Research, vol. 47, no. 2, pp. 295–308,2010.

[11] Q. Yang and Y. Hu, "Application of improved apriori algorithm on educational information," In Proc. of Fifth International Conference on ICGEC

[12] S. Jian-jing, C. Bing, S. Chang-xing, and W. Yun-cheng, "An improvement apriori arithmetic based on rough set theory," In Proc.of PACCS.

[13] H. yu Wang, Xiao-juan, J. Y. Xue, and X. Liu, "Applying fast-apriori algorithm to design data mining engine," In Proc. of International Conference on System Science, Engineering Design and Manufacturing Informatization, vol. 1, Nov 2010.