ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 431- 434 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Frequent Pattern Comparative Analysis of Apriori and FLAG Matrix

[1]Vysyaraju Raisha,[2]V. Sangeetha
[1]Final M.Tech Student, Department of CSE, Pydah College of Engineering and Technology, Visakhapatnam
[2]Associate Professor, Department of CSE,Pydah College of Engineering and Technology,Visakhapatnam
[1]raisha589@gmail.com [2]sangeetaviswanadham@yahoo.com

Abstract: Data mining is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. In this paper we are proposed to find the frequent sequential patterns in a large uncertain database. For the finding of frequent pattern we are using Flag Matrix approach. The Flag Matrix approach is efficient and more flexible for finding the frequent pattern. By using this approach we can reduce the time complexity for finding frequent patterns.

INTRODUCTION

Sequential pattern mining, which discovers frequent subsequences as patterns in a sequence database, is an important data mining problem with broad applications, including the analysis of customer purchase behavior, Web access patterns, scientific experiments, disease treatments, natural disasters, DNA sequences, and so on. The sequential pattern mining problem was first introduced by Agawam and Spirant in Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given user-specified min support threshold, sequential pattern mining is to find all of the frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than min support. Many studies have contributed to the efficient mining of sequential patterns or other frequent patterns in time related data. Almost all of the previously proposed methods for mining sequential patterns and other time-related frequent patterns are based on the property proposed in association mining, which states the fact that any super-pattern of a non frequent pattern cannot be frequent.

The sequence mining function is to get attributes and share them across the database in large content For example, consider the sales database of a bookshop, where the objects resembles customers and the attributes resembles authors or books. Let's consider that the database save each customer record who bought the book in certain period of time, Books which are most frequently bought by the customer are considered as sequence pattern. An example could be that, "70% of the people who buy Jane Austen's Pride and Prejudice also buy Emma within a month." These are used for placement of shelf, promotions etc as patterns. Consider another example of a web access database at a popular site,

Where an object is a web user and an attribute is a web page. The discovered patterns are the sequences of most frequently accessed pages at that site. This kind of information can used to restructure for the website in another way inserting relevant dynamically links in web pages based on user access patterns, Similar type of domains identifying plain failures has been included for the identification of sequence mining(Saki, Lash, & Ogihara, 1998), finding network alarm patterns (Atone et al., 1996), and so on. The task of discovering all frequent sequences in large databases is quite challenging. The search space is extremely large. For example, with m attributes there are $O(m_y)$potentially frequent sequences of length k. With millions of objects in the database the problem of I/O minimization becomes paramount. However, most current algorithms are iterative in nature, requiring as many full database scans as the longest frequent sequence; clearly a very expensive process. Some of the methods, especially those using some form of sampling, can be sensitive to the data-skew, which can adversely affect performance. Furthermore, most approaches use very complicated internal data structures which have poor locality (Parthasarathy, Zaki, & Li, 1998), and add additional space and computation over heads. Our goal is to overcome all of these limitations.

RELATED WORK

The problem of mining frequent sequential patterns (*FSPs*) from deterministic databases has attracted a lot of attention in the research community due to its wide spectrum of real life applications. For example, in mobile tracking systems, Spare useful in the classification or clustering of moving objects and in biological research, FSP mining can help discover correlations among gene sequences .Data uncertainty is inherent in many real-world applications such as sensor data monitoring, RFID localization and location based services, due to environmental factors, device limitations, privacy issues, etc. As a result, uncertain data mining has attracted lot of attention in recent research. A comprehensive survey of the techniques on uncertain data mining can be found in .To our knowledge, is the only work that studies mining sequential patterns from uncertain data. However, this work adopts expected support as the measurement of pattern frequentness, which has some inherent weaknesses with respect to the fundamental probability model, and is therefore ineffective for mining high-quality Sequential patterns from uncertain sequence databases. We will illustrate this point in our later examples.

As one of the most essential tasks in data mining and machine learning area, classification has been studied for many years. Many effective models and algorithms have been proposed to solve the problem in different

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 431- 434 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

aspects, including decision tree, rule-based classifier, support vector machine, etc. Unlike some traditional rule-Tried to mining the complete set off frequent patterns from the input dataset, given the user-specified minimum support threshold and/or discriminative measurements like minimum confidence threshold. Sequential covering technology is further employed to select the most discriminative patterns while covering most input training instances. A test instance misclassified later using classifier trained based on the mined patterns. CBA is one of the most classical associative classification algorithms.

Empirical results show that associative classification algorithm could provide better classification accuracy than other algorithm son categorical datasets. However, this approach takes great amount of running time in both pattern mining and feature selection, since most of the mined frequent patterns are not the most discriminative ones and will be dropped later. To improve the efficiency of associative classification, several algorithms have been proposed in recent years to try to mine the most discriminative patterns directly during the pattern mining step. Different discriminative measures and different instance covering technologies have also been devised. One of the most typical algorithms is HARMONY which uses confidence to evaluate the discrimination of patterns. It employs a so-called instance-centric framework to find one most discriminative pattern for each instance.

In the traditional approach of apriori algorithm it finds the frequent item sets by initially reading the patterns and maintains individual count of the items for frequent one item set. Before generation of the frequent one item set, It generates the candidate set for possible number of items in the transaction table if any pattern which satisfies the minimum threshold value in the candidate set can be treated as frequent item set else not frequent.

For the generation of frequent n item sets, we perform join and prune mechanism for find the possible item sets we use join and to eliminate some item set which are not frequent in previous frequent tem set, then we can generate candidate sets for the next frequent item sets which satisfies the minimum threshold value or support count But the main drawbacks with apriori algorithm is multiple candidate set generation and multiple database scan.

For every iteration candidate set should be generated before retrieving frequent item set, when number of items increases automatically complexity of the pattern generation process increases, time complexity is more with apriori algorithm, one more major drawback is multiple database scans, to check the items et is frequent or not we need to scan database for every iteration. Recently, due to the wide application of uncertain data, such as sensor network monitoring , moving object search , object identification, etc., mining frequent item sets over uncertain (or probabilistic) data has attracted much attention from the data mining and database communities. However, like its counterpart, mining frequent item sets over exact data, mining frequent item sets over uncertain data can produce an based algorithms like Ripper or FOIL, it

exponential number of frequent item sets which cause the mining results less useful. A classical solution is proposed to address the similar problem in exact data, called mining frequent closed item sets. The approach intends to find frequent item sets whose supports are different from the supports of all their supersets; thus, redundant item sets will be excluded. Since mining frequent closed item sets offers high-level compression, in this paper, we would like to investigate this compression strategy on uncertain data in order to reduce the huge number of frequent item sets generated by the previous mining methods.

PROPOSED SYSTEM

The proposed technique is used to find out patterns by using the data mining technique. In this paper we are proposing new technique for finding frequent patterns i.e. Flag Matrix approach. By using this approach we find the frequent patterns and also reduce the time complexity for finding frequent items. Thus the research presented a new algorithm of mining maximum frequent item sets first based on the Flag Matrix of frequent length-1 item sets. The main idea of the algorithm is to create a Flag Matrix with frequent length-1 item sets as row headings and transaction records' IDs as column headings (Fig 4). In the matrix, there are only two types of values, '1' and '0', which means that the transaction record contains or not the corresponding frequent length-1 item set.

Then it is necessary to calculate the number of value 1 in each column and the count of the columns with the same number of value 1. If the count of those columns is larger than the minimum support, in accordance, the number of value 1 in the column may be the size of maximum frequent item set, vice versa. Therefore, some values of which each may be the maximum frequent item set's length will be calculated. Subsequently, a set of candidate item sets used for extracting maximum frequent item sets will be generated from frequent length-1 item sets according to each maximum value and the support of each candidate item set will be calculated based on the Flag Matrix. If the support is larger than the minimum support, the Candidate item set is frequent, vice versa. Finally, all the frequent item sets will be extracted from maximum frequent item sets according to the nonempty sub-sets of frequent item sets being still frequent. Generally speaking, the main principles of the new algorithm include three aspects:

Flag Matrix

The server or service provider performing association rule mining on cipher database for finding maximum frequent item sets. Thus the research presented a new algorithm of mining maximum frequent item sets first based on the Flag Matrix of frequent length-1 item sets. The main idea of the algorithm is to create a Flag Matrix with frequent length-1 item sets as row headings and transaction records' IDs as column headings In the matrix, there are only two type of values, '1' and '0', which means that the transaction record contains or not the corresponding frequent length-1 item set. Then it is necessary to calculate the number of value

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 431- 434 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

1 in each column and the count of the columns with the same number of value 1

Algorithm for Flag Matrix construction:

Step1: While e (true) // patterns available

Step2: Read the individual pattern P_i separated by a special character.

Step3: Construct an empty matrix with I rows and j columns
Where 'I' is item and 'j ' is transaction id

Algorithm for frequent pattern Generation from Flag Matrix:

Step1: read item set {I_1,I_2…In) and Initialize counter:=0 ,final counter :0

Step2: for i:=0 ;i< n ;i++

For j:=0 j<trans _size ;j++

If intersection of (i,j)==1
Counter :+1;

Next

Flag Matrix can be constructed based on the availability of the item with respect to transaction. Initially the first transaction contains "a,b,c,d" ,So in corresponding positions of items set to '1' in first transaction
else '0' and consider second transaction "a,c,e", set the corresponding item positions to

Step4: Set intersection (imp)=1 if corresponding g item 'I' available in particular transaction ID 'J' .else set to 0.

Step5: Continue step 2 to 5

Now we can extract frequent patterns from the matrix, to extract frequent 1 item set, initially count number of ones in vertical columns with respect to item, if it matches minimum threshold values then treat it as frequent item else ignore, continue same process for 2 item set check whether two items have '1' in their corresponding vertical columns then increment, continue until all transactions verified. If total count greater than threshold value then treat it as frequent item
If counter ==I_i.size() then add to item list
Next
Step3: Set minimum threshold value (t)

Step4: for k=0;k<itemlist_size ;k++

If item_list[k].count >= t Then

add to frequent item list
End if
Next

Step5: return frequent pattern list

'1' in second transaction, continue the process until all transactions get completed.

Itemset	Transaction IDS					
	1	2	3	4	5	6
a	1	1	0	0	0	1
b	1	0	1	1	0	1
c	1	1	0	1	1	1
d	1	0	1	0	1	1
e	0	1	0	1	1	1

Fig 4: Flag Matrix

Frequent patterns generation

Initially frequent one item set can be generated by counting number of individual items in all transactions like, Consider item 'a', now count number of '1's opposite to item 'a' in all transactions, total count of a is 3 because a available in transaction1 2 and 6.if the count equal or greater than minimum threshold value or support count (2 in our example) it can be treated as frequent item.

Frequent 'n' item set

To find the frequent two item set or three item set or n item set, we can follow the same procedure until frequent items found. Consider two item set {a,b},now check the corresponding ones opposite to "a,b" (both should be set to "1"),then count would be "1".In the above table transaction 1 and 6 contains "1" in both

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 431- 434 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

places of a and b, so count is 2. Now {a,b} is a frequent item ,because our minim support count value is 2,by the same process you can find the remaining frequent patterns.

CONCLUSION

We are concluding our research work with efficient frequent pattern mining approach with comparative analysis of traditional Apriori and Flag matrix, the experimental analysis of flag matrix shows efficient results than Apriori in terms of time complexity.

REFERENCES

1. J. Pei *et al*., "PrefixSpan: Mining sequential patterns efficientlyby prefix-projected pattern growth," in *Proc. 17th ICDE*, Berlin,Germany, 2001

2.M. J. Zaki, "SPADE: An efficient algorithm for mining frequentsequences," *Mach.Learn.*, vol. 42, no. 1–2, pp. 31–60, 2001.

3. Z. Zhao, D. Yan, and W. Ng, "Mining probabilistically frequentsequential patterns in uncertain databases," in *Proc. 15th Int. Conf.EDBT*, New York, NY, USA, 2012.

4. C. Gao and J. Wang, "Direct mining of discriminative patternsfor classifying uncertain data," in *Proc. 16th ACM SIGKDD*,Washington, DC, USA, 2010.

5. Y. Tong, L. Chen, and B. Ding, "Discovering threshold-based frequentclosed itemsets over probabilistic data," in *Proc. IEEE 28thICDE*, Washington, DC, USA, 2012.

6. L. Wang, R. Cheng, D. Lee, and D. Cheung, "Accelerating probabilisticfrequent itemset mining: A model-based approach," in*Proc. 19th ACM CIKM*, Toronto, ON, Canada, 2010.

7. J. Brickell and V. Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In ASIACRYPT, pages 236–252, 2005.

8. D.W.L. Cheung, J. Han, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In PDIS, pages 31–
42, 1996.

9. D.W.L Cheung, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. Efficient miningof association rules in distributed databases. IEEE Trans. Knowl. Data Eng., 8(6):911–922, 1996.

10. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31:469–
472, 1985