www.ijecs.in

International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 9 Sep 2015, Page No. 14371-14377

A Study On Association Rule Mining

S.Venkata Krishna Kumar[1] *P.Kiruthika*[2]
Associative Professor, Department of Computer Science,
PSG College of Arts & Science, Coimbatore,
Tamilnadu, India

Research Scholar, Department of Computer Science,
Dr.N.G.P Arts & Science college, Coimbatore,
Tamilnadu, India

Abstract: Now a days, Association Rule Mining become a major Role in Data mining. It attracts more attention because of its wide applicability. Association Mining aims to extract interesting correlations, frequent patterns, associations or casual structure among sets of items. This paper aims at giving a theoretical study on some of the existing algorithms. The concepts behind association rules are provided at the beginning followed by an overview to some of the previous research works done on this area. The merits and demerits are discussed and concluded with inference.

Keywords: Data Mining, Association rule, frequent item sets,AIS,SETM,Apriori, Aprioritid, apriorihybrid, FP-Growth Algorithm

I INTRODUCTION

Data mining is generally thought of as the process of finding hidden, non trivial and previously unknown information in large collection of data. Association rule Mining is an important component of data mining. Association rule are an important class of methods of finding pattern in data.KDD is the process of identifying a valid, potentially useful and ultimately understandable structure in data. This process involves selecting or sampling data from a data warehouse, Cleaning or preprocessing it, transforming or reducing it, applying a data mining component to produce a structure, and then evaluating the derived structure. Data mining is step in the KDD Process concerned with the algorithmic means by which patterns or structures are enumerated from the data under acceptable Computational efficiency limitations.

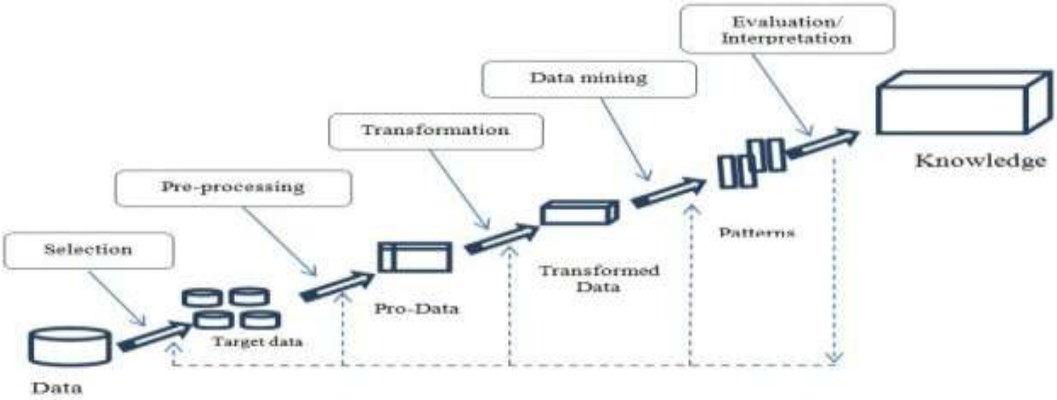

Figure 1: Steps in Data mining

Data cleaning and integration is the process of removing noise and inconsistent data, combining data from multiple sources. Data Selection is the process of retrieving relevant data from multiple sources. Data Transformation is the process, data are transformed or consolidate in to forms appropriate for mining by performing summary or aggregation operation. Data miming is an essential process where intelligent methods are applied in order to extract data patterns. Knowledge presentation is visualization and knowledge representation techniques are used to present the mined knowledge to the user.

1.1 Types of Data Mining

Data mining tasks are classified into two categories Predictive mining and Descriptive mining. Predictive mining tasks perform inference on the current data in order to make predictions. Descriptive mining tasks characterize the general properties of the data in the data base.

II ASSOCIATION RULE MINING

Association rules are widely used in various areas such as telecommunication networks, market and Risk Management, inventory control, cross-marketing, catalog design, Clustering and Classification. Association rules measured by Confident and Support. Support is the percentage of transaction in D that contains AUB. Confident is the percentage of transactions in D Containing A that also contain B. Support (A⇒B) =P(AUB), confident (A⇒B) =P(B/A). Rules that satisfy both a minimum support threshold (min_sup) and minimum confident threshold (min_conf) are called Strong. For example

Transaction ID	Item Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Table 1

98% of people who purchase tires and auto accessories also get automotive services done. Let Minimum support 50%, and minimum Confident 50%, we have A⇒C (50%, 66.6%), C⇒A (50%,100%).In general, association rule mining can be viewed as a two step process the first step is find all Frequent item sets and each of these item sets will occur at least as frequently as a predetermined minimum support count, min_sup. The second step is generating strong association rules from the frequent item sets, these rules must satisfy minimum support and minimum confidence.

III AIS ALGORITHM

It deals on improving the quality of databases together with necessary functionality to process decision support queries. This was the first algorithm used for mining association rules. AIS (Agrawal, Imielinski and Swami) algorithm consists of phases. Frequent item sets were generated in the first phase and the second phase deals on generation of confident and frequent association rules. In this algorithm only one item consequent association rule are generated for example rules like A n B ⇒C can be generated but not the rules like A⇒B n C. The setback of the AIS algorithm is it generates multiple phases over the databases and counts lot of candidate item sets that are too small, which needs more space and waste lot of effort that are not useful. To make more efficient of this algorithm an estimation method was introduced to avoid those candidate item sets that are too small and unnecessary effort of counting those item sets. Since all the candidate item sets are usually store in the main memory, memory management is also proposed when memory is not enough.

TID	ITEMS
100	1,2,4
200	2,3,5
300	1,3,2,5
400	2,5

ITEMSET	SUPPORT
{1}	2
{2}	3
{3}	3
{4}	3

C1

ITEMSET	TID
{1,3}	100
{1,4}	100
{2,4}	100
{2,3}	200
{2,5}	200
{3,5}	200
{1,2}	300
{1,3}	300
{1,5}	300
{2,3}	300
{2,5}	300
{3,5}	300
{2,5}	400

ITEMSET	SUPPORT
{1,3,4}	1
{2,3,5}	2
{1,3,5}	1

C2

Figure 2: Example of AIS

IV SETM ALGORITHM

The SETM algorithm was motivated to compute large item sets using SQL. In this algorithm candidate item sets generated on the fly as the databases scanned but counted at end of the pass. It does the same work as the AIS does. However to do the standard SQL operation for candidate generation SETM algorithm separates candidate generation from counting but the transaction identifier TID of the generating transaction is saved with the candidate item sets in a sequential structure. At the end of the pass the support count of the candidate item set is determined by sorting and aggregating this sequential structure.

Database

TID	ITEMS
100	1,2,4
200	2,3,5
300	1,3,2,5
400	2,5

C1

ITEMSET	SUPPORT
{1}	2
{2}	3
{3}	3
{4}	3

ITEMSET	Support
{1,3}	2
{1,4}	1
{2,4}	1
{2,3}	2
{2,5}	3
{3,5}	2
{1,2}	1
{1,5}	1

ITEMSET	TID
{1,3,4}	100
{2,3,5}	200
{1,3,5}	300
{2,3,5}	300

C2

Figure 3: Example of SETM

V APRIORI ALGORITHM

Apriori algorithm is mainly used for finding frequent item set in given data items. The algorithm use a level-wise search, where K-item sets are used to explore(K+1)item sets, to mine frequent item sets from transactional database for Boolean association rules. The frequent subsets are extended one item at a time and this step is known as candidate generation process that candidates are tested against the data. To count candidate item sets efficiently, apriori uses breath-first search method and a hash tree structure.

This method identifies the frequent individual items in the database and extends them to larger and larger item sets as long as those item sets appear sufficiently often in the database. Apriori algorithm determines frequent item sets that can be used to determine association rules.

Following is the procedure for Apriori Algorithm:

```
$CI_k$: Candidate item set having size K
$FI_k$: Frequent item set having size K
$FI_1$: {frequent items}
For(k=1; $FI_k$!=null; K++)do begin
$CI_{k+1}$: Candidates generated from $FI_k$;
For each transaction t in database D do Increment the count value of all candidates in $CI_{k+1}$ that are contained in t
$FI_{k+1}$ = Candidates in $CI_{k+1}$ with min_support End Return $FI_k$
```

TID	ITEMS
100	1,2,4
200	2,3,5
300	1,3,2,5
400	2,5

Database

ITEMSET	SUPPORT
{1}	2
{2}	3
{3}	3
{4}	3

C1

ITEMSET	SUPPORT
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

C2

ITEMSET	SUPPORT
{2,3,5}	2

Figure 4: Example of Apriori

There are two drawbacks of the apriori algorithm. First is the complex candidate generation process which uses most of the time, space and memory. Another drawback is it requires multiple scans of the database.

VI APRIORITID ALGORITHM

This algorithm also uses the apriori generation function to determine the candidate item set before the pass begins. The interesting feature of this algorithm is that database is not used for counting the support of candidate item set after the first pass. It not necessary to use that same algorithm in all the passes over the data. Apriori examine every transaction in the database rather than scanning the databases, Aprioritid scans C_k for obtaining support counts and the size of C_k is smaller than size of database. Based on this observation apriorihybrid algorithm has been designed.This uses apriori in the initially passes and switch to aprioritid in later passes. The main merit of this algorithm is that the later passes the performance of Aprioritid is better than Apriori.

TID	ITEMS
100	1,2,4
200	2,3,5
300	1,3,2,5
400	2,5

Database

TID	ITEMS
100	1,3,4
200	2,3,5
300	1,3,2,5
400	2,5

C1

ITEMSET	Support
{1}	2
{2}	3
{3}	3
{5}	3

L1

ITEMSET	SUPPORT
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

C2

DOI: 10.18535/ijecs/v4i9.52

TID	Support
100	{1,3}
200	{2,3},{2,5},{3,5}
300	{1,2},{1,3},{1,5},{2,3},{2,5},{3,5}
400	{2,5}

ITEMSET	SUPPORT
{1,3}	2
{2,3}	2
{2,5}	3
{3,5}	2

ITEMSET	Support
{2,3,5}	2

C3

TID	ITEMS
200	{2,3,5}
300	{2,3,5}

ITEMSET	Support
{2,3,5}	2

Figure 5: Example of AprioriTid

VI APRIORIHYBRID ALGORITHM

Apriori and aprioritid use the same candidate generation procedure and therefore count the same item sets the later passes the number of candidate item sets reduces. However, uses apriori in initial passes and switches as to aprioritid in later. So based on this observation new algorithm is designed that is apriorihybird which uses features of both above algothms. It helps using apriori algorithm in earlier passes aprioritid algorithm in later passes.

VII FP-GROWTH ALGORITHM

To break the two drawbacks of Apriori algorithm, FP-growth algorithm is used. FP-growth requires constructing FP-tree. For that, it requires two passes. FP-growth uses divide and conquer strategy. It requires two scans on the database. It first computes a list of frequent items sorted by frequency in descending order(F-list)and during its first database scan. In the second scan, the database is compressed in to FP-tree. This algorithm performs mining on FP-tree recursively. There is a problem of finding frequent item sets which is converted to searching and constructing trees recursively. The frequent item sets are generated with only two passes over the database and without any candidate generation process. There are two sub processes of frequent patterns generation process which includes: construction of the FP-tree, and generation of the frequent patterns from the FP-tree.

FP-tree is constructed over the data-set using 2 passes are as follows:

Pass1:

1)Scan the data and find support for each item.

2)Discard infrequent items.

3)Sort frequent items in descending order which is based on their support.

By using this order we can build FP=tree, so that common prefixes can be shared.

Pass2:

1)Here nodes correspond to items and it has a counter

2) FP-growth reads one transaction at a time and then maps it to a path.

3) Fixed order is used, so that paths can overlap when transactions share the items.

In this case, counters are incremented. Some pointers are maintained between nodes which contain the same item, by creating singly linked lists. The more paths that overlap, higher the compression. FP-tree fit in memory. Finally, frequent item sets are extracted from the FP-Tree.

COMPARISON OF ASSOCIATION RULE MINING ALGORITHMS

characteristics	**AIS**	**SETM**	**Apriori**	**Aprioritid**	**Apriorihybrid**	**FP-growth**
Data support	Less	Less	Limited	Often suppose large	Very large	Very large
Speed in initial phase	Slow	Slow	High	Slow	High	High
Speed in later phase	Slow	slow	Slow	high	High	High
Accuracy	Very less	Less	Less	More accurate than Apriori	More accurate than Aprioritid	More accurate

CONCLUSION

There are many number of association rule algorithm available but this paper represents comparison of six association rule mining algorithm which are AIS,SETM,Apriori,Aprioritid,Apriorihybrid and FP growth. Comparison is done based on the above performance criteria each algorithm has some merits and demerits from the above comparison. We came to a that FP-growth performs better than all the other algorithms we mentioned here.

REFERENCES

1. Aggarwal, R., and Srikant, R. “Fast Algorithms for Mining Association Rules”. Proceedings of the 20th VLDB Conference Santiago, Chile, 1994
2. Webb, I.G. “Efficient Search for Association Rules”. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000), Boston, MA, New York, NY
3. Hipp, J., G¨untzer, U., and Nakhaeizadeh, G. “Algorithms for Association Rule Mining – A General Survey and Comparison”, SIGKDD Explorations ACM, JULY 2000.
4. Hunyadi, D.“Performance comparison of Apriori and FP-Growth Algorithms in Generating Association Rules”. Proceedings of the European Computing Conference ISBN: 978-960-474-297-4.
5. Borgelt, C. **“**Efficient Implementations of Apriori and Eclat”. Workshop of frequent item set mining implementations (FIMI 2003, Melbourne, FL, USA).