Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 6, June 2015, pg.271 – 277

RESEARCH ARTICLE

Association Rule Mining: Algorithms Used

Himani Bathla, Ms. Kavita Kathuria

M.Tech student, Department of CSE, Shri Baba Masth Nath Engineering College

Assistant professor, Department of CSE, Shri Baba Masth Nath Engineering College

himanibathla989@gmail.com, kavita.kathuria1990@gmail.com

***Abstract:** Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. Data mining as a synonym for another popularly used term, Knowledge Discovery from Data, or KDD. Mining Associations is one of the techniques involved in the process mentioned in chapter 1 and among the data mining problems it might be the most studied ones. Discovering association rules is at the heart of data mining. Mining for association rules between items in large database of sales transactions has been recognized as an important area of database research.*

***Keywords:** KDD, WWW, CAR, CHAID, AIS*

I. INTRODUCTION

Data mining, also known as knowledge discovery in databases, is such a research area to extract implicit, understandable, previously unknown and potentially useful information from data. Data mining helps to extract important data from a large database. It is the process of sorting through large amounts of data and picking out relevant information through the use of certain sophisticated algorithms. As more data is gathered, with the amount of data doubling every three years, data mining is becoming an increasingly important tool to transform this data into information. Data mining as a synonym for another popularly used term, Knowledge Discovery from Data, or KDD. Alternatively, others view data mining as simply an essential step in the process of knowledge discovery. Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time.

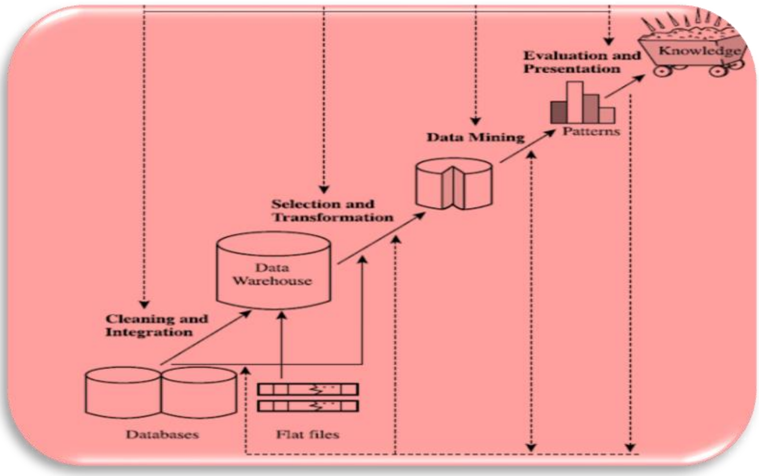

Figure 1.1: Data mining as a step in the process of knowledge discovery

Knowledge discovery as a process is depicted in Figure 1.1 and consists of an iterative sequence of the following steps:

- Data cleaning (to remove noise and inconsistent data)
- Data integration (where multiple data sources may be combined)
- Data selection (where data relevant to the analysis task are retrieved from the database)
- Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)
- Data mining (an essential process where intelligent methods are applied in order to extract data patterns)
- Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures
- Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

II. ARCHITECTURE AND SCOPE OF DATA MINING

Data mining, also known as knowledge discovery in databases, is such a research area to extract implicit, understandable, previously unknown and potentially useful information from data. Data mining helps to extract important data from a large database. Data mining is the process of discovering interesting knowledge from large amounts of data stored in databases, data warehouses, or other information repositories. Based on this view, the architecture of a typical data mining system may have the following major components:

- **Database, data warehouse, WorldWideWeb, or other information repository:** This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.
- **Database or data warehouse server:** The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.

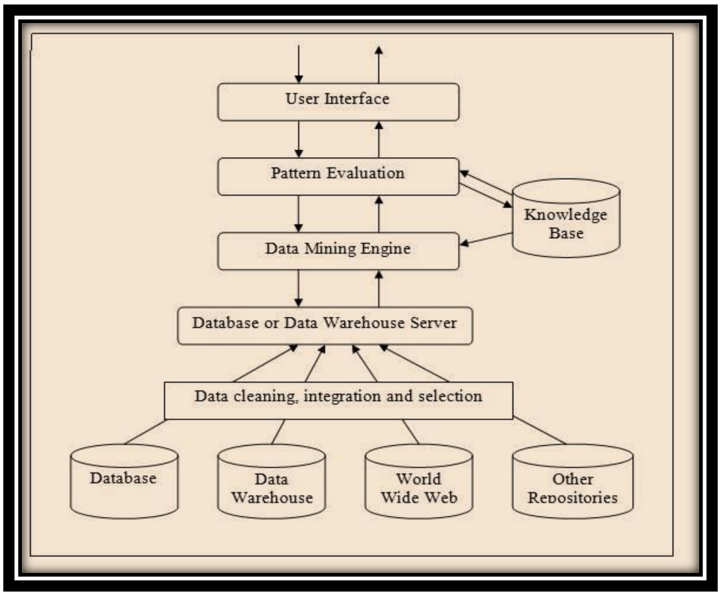

Figure 1.2 : Architecture of Data Mining

- **Knowledge base:** This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).
- **Data Mining Engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.
- **Pattern Evaluation Module:** This component typically employs interestingness measure and interacts with the data mining modules so as to *focus* the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.
- **User interface:** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

III. ASSOCIATION RULE MINING: INTRODUCTION

i) Mining for association rules between items in large database of sales transactions has been recognized as an important area of database research. The original problem addressed by association rule mining was to find a correlation among sales of different products from the analysis of a large set of super market data. Mining Associations is one of the techniques involved in the process mentioned in chapter 1 and among the data mining problems it might be the most studied ones.

In general, association rule mining can be viewed as a two-step process:

i) **Find all frequent itemsets:** By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, *min sup*.

ii) **Generate strong association rules from the frequent itemsets:** By definition, these rules must satisfy minimum support and minimum confidence.

A major challenge in mining frequent itemsets from a large data set is the fact that such mining often generates a huge number of itemsets satisfying the minimum support (*min sup*) threshold, especially when *min sup* is set low. This is because if an itemset is frequent, each of its subsets is frequent as well.

An itemset *X* is **closed** in a data set *S* if there exists no proper super-itemset *Y* such that *Y* has the same support count as *X* in *S*. An itemset *X* is a **closed frequent itemset** in set *S* if *X* is both closed and frequent in *S*. An itemset *X* is a **maximal frequent itemset (or max-itemset)** in set *S* if *X* is frequent, and there exists no super-itemset *Y* such that *X* *Y* and *Y* is frequent in *S*.

ii) Frequent Pattern Mining: Market basket analysis is just one form of frequent pattern mining. Frequent pattern mining can be classified in various ways, based on the following criteria:

- **Based on the *completeness* of patterns to be mined:** We can also mine constrained frequent itemsets (i.e., those that satisfy a set of user-defined constraints), approximate frequent itemsets (i.e., those that derive only approximate support counts for the mined frequent itemsets), near-match frequent itemsets (i.e., those that tally the support count of the near or almost matching itemsets), top-*k* frequent itemsets (i.e., the *k* most frequent itemsets for a user-specified value, *k*), and so on.
- **Based on the *levels of abstraction* involved in the rule set:** Some methods for association rule mining can find rules at differing levels of abstraction. For example, suppose that a set of association rules mined includes the following rules where *X* is a variable representing a customer:

buys(*X*, "*computer*") => *buys*(*X*, "*HP printer*")

buys(*X*, "*laptop computer*") => *buys*(*X*, "*HP printer*") .

In Above rules the items bought are referenced at different levels of abstraction (e.g., "*computer*" is a higher-level abstraction of "*laptop computer*").

- **Based on the number of data dimensions involved in the rule:** If the items or attributes in an association rule reference only one dimension, then it is a single-dimensional association rule.

buys(*X*, "*computer*") => *buys*(*X*, "*antivirus software*")

- **Based on the types of values handled in the rule:** If a rule involves associations between the presence or absence of items, it is a Boolean association rule.
- **Based on the kinds of rules to be mined:** Frequent pattern analysis can generate various kinds of rules and other interesting relationships. Association rules are the most popular kind of rules generated from frequent patterns. Typically, such mining can generate a large number of rules, many of which are redundant or do not indicate a correlation relationship among itemsets. Thus, the discovered associations can be further analyzed to uncover statistical correlations, leading to correlation rules.

IV. ASSOCIATION RULE MINING: ALGORITHM

Most algorithms used to identify large itemsets can be classified as either sequential or parallel.

i) Sequential Algorithms:

i) **AIS:** The AIS algorithm makes multiple passes over the entire database. During each pass, it scans all transactions. In the first pass, it counts the support of individual items and determines which of them are large or frequent in the database. Large itemsets of each pass are extended to generate candidate itemsets. After scanning a transaction, the common itemsets between large itemsets of the previous pass and items of this transaction are determined. The AIS algorithm was the first published algorithm developed to generate all large itemsets in a transaction database[17]. It focused on the enhancement of databases with necessary functionality to process decision support queries. This algorithm was targeted to discover qualitative rules. This technique is limited to only one item in the consequent.

Advantage:

- The algorithm was used to find if there was an association between departments in the customer's purchasing behavior.

Disadvantage:

- The main problem of the AIS algorithm is that it generates too many candidates that later turn out to be small
- Besides the single consequent in the rule, another drawback of the AIS algorithm is that the data structures

  required for maintaining large and candidate itemsets were not specified.

ii)**SETM:** Similar to the AIS algorithm, the SETM algorithm makes multiple passes over the database. In the first pass, it counts the support of individual items and determines which of them are large or frequent in the database. Then, it generates the candidate itemsets by extending large itemsets of the previous pass. In addition , the SETM remembers the TIDs of the generating transactions with the candidate itemsets. The relational merge-join operation can be used to generate candidate itemsets.

Advantage:

- Generating candidate sets, the SETM algorithm saves a copy of the candidate itemsets together with TID of the generating transaction in a sequential manner.

Disadvantage:

- Since for each candidate itemset there is a TID associated with it, it requires more space to store a large number of TIDs.
- SETM is not efficient and there are no results reported on running it against a relational DBMS.

iii) **APRIORI:** It is by far the most well-known association rule algorithm. The fundamental differences of this algorithm from the AIS and SETM algorithms are the way of generating candidate itemsets and the selection of candidate itemsets for counting. The Apriori generates the candidate itemsets by joining the large itemsets of the previous pass and deleting those subsets which are small in the previous pass without considering the transactions in the database. By only considering large itemsets of the previous pass, the number of candidate large itemsets is significantly reduced. The apriori_gen() function as described in[1] has two steps.

- During the first step, L_{k-1} is joined with itself to obtain Ck.

- In the second step, apriori_gen() deletes all itemsets from the join result, which have some (k-1)–subset that is not in L_{k-1}. Then, it returns the remaining large k-itemsets.

Large Itemsets in the third pass (L_3)	Join (L_3, L_3)	Candidate sets of the fourth pass (C_4 after pruning)
{{Apple, Bagel, Chicken}, {Apple, Bagel, DietCoke}, {Apple, Chicken, DietCoke}	{{Apple, Bagel, Chicken, DietCoke}, {Apple, Chicken, DietCoke Eggs}}	{{Apple, Bagel, Chicken, DietCoke}}

Table 1 Finding Candidate Sets Using Apriori_gen()

Apriori incorporates buffer management to handle the fact that all the large itemsets Lk-1 and the candidate itemsets Ck need to be stored in the candidate generation phase of a pass k may not fit in the memory. A similar problem may arise during the counting phase where storage for Ck and at least one page to buffer the database transactions are needed[1]. [1] considered two approaches to handle these issues. At first they assumed that Lk-1 fits in memory but Ck does not. The performance of Apriori was assessed by conducting several experiments for discovering large itemsets on an IBM RS/6000 530 H workstation with the CPU clock rate of 33 MHz, 64 MB of main memory, and running AIX 3.2. Experimental results show that the Apriori algorithm always outperforms both AIS and SETM [1].

Algorithm 3 shows the Apriori technique. As mentioned earlier, the algorithm proceeds iteratively.

Function count(C: a set of itemsets, D: database)

Algorithm 3. Apriori [1]

Input:

I, D, s

Output:

L

Algorithm:

//Apriori Algorithm proposed by Agrawal R., Srikant, R. [**1**]

//procedure LargeItemsets

```
1) C 1: = I; //Candidate 1-itemsets
2) Generate L1 by traversing database and counting each occurrence of an attribute in a transaction;
3) for (k = 2; $L_{k-1} \neq \varphi$; k++) do begin
//Candidate Itemset generation
//New k-candidate itemsets are generated from (k-1)-large itemsets
4) Ck = apriori-gen($L_{k-1}$);
//Counting support of Ck
5) Count (Ck, D)
6) Lk = {c    $C_k$ | c.count ≥ minsup}
7) end
9) L =  k$L_k$
```

V. CONCLUSION

In this PAPER we have discussed various association rule algorithms and compared two algorithms: Apriori algorithm and Filter Associator. We have analyzed the frequent itemsets generation and number of cycle performed over the Apriori algorithm and Filter Associator in the context of association analysis. According to the comparison of above two algorithms on weka tool, we conclude that Filter Associator is efficient algorithm than Apriori algorithm based on above two factors (Number of cycle performed, large itemsets) because the Apriori algorithm generates more number of cycle performed and generate extra large itemsets which degrades the performance of algorithm.

REFERENCES

1. Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In Proc. 20th Int. Conf. Very Large Data Bases, 487-499.

2. A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. Proceedings of the 21st International Conference on Very large Database,1995.

3. Anurag Choubey, Ravindra Patel, J.L.Rana, “A Survey of Efficient Algorithms and New Approach for Fast Discovery of Frequent itemset for Association Rule Mining”, IJSCE ,ISSN: 2231-2307, vol. 1, issue 2,May 2011.

4. Dr. Varun Kumar, Anupama Chadha, “Mining Association Rules in Student’s Assessment Data”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012.

5. Du Ping, Gao Yongping, “ A New Improvement of Apriori Algorithm for Mining Association Rules”, International Conference on Computer Application and System Modeling (ICCASM 2010), Vol. 2, 529-532.

6. Farah Hanna AL-Zawaidah, Yosef Hasan Jbara, “An Improved Algorithm for Mining Association Rules in Large Databases”, World of Computer Science and Information Technology Journal (WCSIT), ISSN: 2221-0741 Vol. 1, No. 7, 311-316, 2011.

7. Hassan M. Najadat, Mohammed Al-Maolegi, Bassam Arkok, “An Improved Apriori Algorithm for Association Rules”, International Research Journal of Computer Science and Application Vol. 1, No. 1, June 2013, PP: 01 – 08.