Implementation of Association Rule Mining using Reverse Apriori Algorithmic Approach

Ashma Chawla
Faculty of CSE Department
Chitkara University, Punjab
Punjab, India

Kanwalvir Singh Dhindsa
Faculty of CSE and IT Department
BBSBEC, Fatehgarh Sahib
Punjab, India

ABSTRACT

Association rule mining is always considered to be the most important task for mining data in almost every field. There have been many algorithms devised for mining frequent patterns till today. Algorithm evolutions started with AIS which was soon upgraded and named as Apriori. Apriori is most widely used algorithm in terms of data mining. In this paper an improved approach to Apriori termed as reverse Apriori is proposed and the results are compared with the classical approach.

General Terms

Algorithms, Data Mining

Keywords

Data Mining, Frequent Pattern Matching, Apriori Algorithm.

1. INTRODUCTION

1.1 Data Mining

Data mining has been proved as a very basic tool in knowledge discovery and decision making process. Data mining technologies are very frequently used in a variety of applications. Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters. Frequent patterns are the itemsets that are frequently visited in database transactions at least for the user defined number of times which is known as support threshold. Presently, a number of algorithms have been proposed in literature to enhance the performance of Apriori Algorithm, for the purpose of determining the frequent pattern. The major concern for any algorithm is to reduce the processing time. Knowledge Discovery in Databases (KDD) and Data Mining (DM) helps to extract useful information from raw data. Association rules describe how often items are purchased together. Such rules can be useful for decisions concerning product pricing, promotions, store layout and many others.

In Figure 1, it is depicted that the data is collected from several sources. It is integrated and placed in some common data store. Part of it is then taken and pre-processed into a standard format. This 'prepared data' is then passed to a data mining algorithm which produces an output in the form of rules or some other kind of 'patterns' [8].

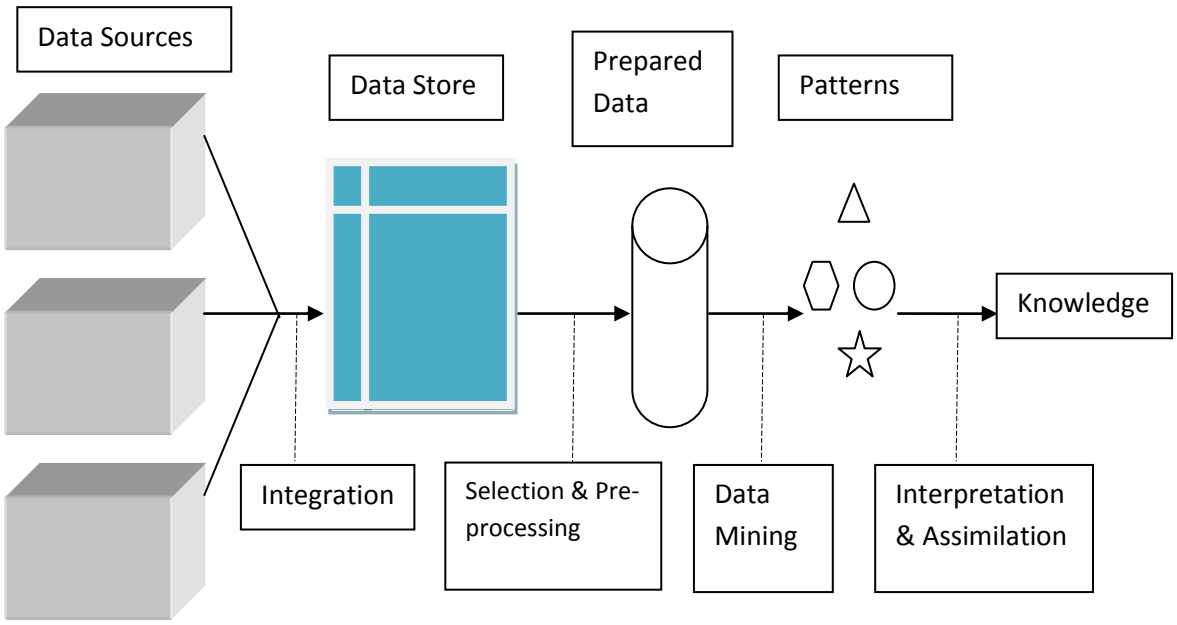

Figure 1: The Knowledge Discovery Process [8]

1.2 Frequent pattern mining

Frequent patterns are patterns that occur frequently in data. There are many kinds of frequent patterns, including item sets, subsequences and substructures. An example of a rule, mined from any transactional database is Buys (X,"Computer") => buys (X,"Software") [support=1%, confidence=50%] where X is a variable representing a customer. A confidence, or certainty, of 50% means that if a customer buys a computer, there is a 50% chance that he will buy software as well. A 1% support means that 1% of all of the transactions under analysis showed that this rule stays. The above rule can be simply written as "computer=>software [1%, 50%]".

2. EXISTING TECHNIQUES AND ALGORITHMS FOR ASSOCIATION RULES

2.1 FP Growth

Frequent pattern growth a very popular association rule mining algorithm for discovering itemsets in a database. The algorithm follows two step approaches for finding interesting rules. The Step1 of the algorithm builds a tree known as FP tree and in step2 frequent items are extracted from this FP tree. FP Growth algorithm is a 2-pass algorithm over database. Where one side FP Growth does not generates any candidate sets and thereby it is considered to be fastest than Apriori, the other side the drawbacks with this algorithm comes up in the form of expensive tree building and the uncertainty of fitting FP tree in memory.

2.2 Apriori

Apriori Algorithm is a decisive algorithm for mining frequent itemsets for Boolean association rules. It uses prior knowledge of frequent itemset properties. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1) itemsets. First the set of frequent 1 itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L1. Next L1, is used to find L2, the set of frequent 2- itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database. Many improved algorithm for Apriori such as AprioriTID, Apriori Hybri, Multiple oins, Reorder and Direct etc have came into existence. The main idea of these algorithms conceptually is that the subset of frequent items is a frequent set and the superset of a infrequent set is an infrequent itemset. Apriori, while historically significant, suffers from a number of inefficiencies. Candidate generation generates large numbers of subsets - the algorithm attempts to load up the candidate set with as many as possible before each scan.

3. LITERATURE REVIEW

Kumar et al. [1] implements three phases of Web usage mining namely preprocessing, pattern discovery, and pattern analysis. Apriori algorithm is used to generate an association rule that associates the usage pattern of the clients for a particular website. The output of the system was in terms of memory usage and speed of producing association rules. A clustering algorithm to find out data clusters for both numerical and nominal data is proposed by Sharma et al. [2] by calculating the average and log values of data set. This algorithm improves the techniques of Web Usage Mining by first discover the log files of individual users at one place.

Martinez-Romo et al. [3] have analyzed different information retrieval methods for both, the selection of terms used to construct the queries submitted to the search engine, and the ranking of the candidate pages that it provides, in order to help the user to find the best replacement for a broken link. To test the sources, they have also defined an evaluation methodology which does not require the user judgments, what increases the objectivity of the results.

A new reactive session reconstruction method is given by Dohare et al. [4]. This algorithm is better than previously developed both time and navigation oriented heuristics as it does not allow page sequences with any unrelated consecutive requests to be in the same session. They have also implemented agent simulator for generating real user sessions. Das et al. [5] analyzed the web server user access logs of Firat University to help system administrator and Web designer to improve their system by determining occurred system errors, corrupted and broken links by using web using mining.

Fayyad et al. [6] have focused on web log file format, its type and location. Log files usually contain noisy and ambiguous data. Preprocessing involves removal of unnecessary data from log file. Data preprocessing is an important step to filter and organize appropriate information before using to web mining algorithm. They have also proposed two algorithms for field extraction and data cleaning. Preprocessing web log file is used in data mining techniques, also used in intrusion detection system as input to detect intrusion. Apriori - the first scalable algorithm designed for association-rule mining algorithm. Apriori is an improvement over the AIS and SETM algorithms stated, Das et al. [7]. The algorithm is based on the large itemset property which states: Any subset of a large itemset is large and if an itemset is not large and then none of its supersets are large.

4. PROPOSED APPROACH

The Association Rule Mining Apriori Algorithm has few drawbacks such as; the iterations involved reduce the minimum support until it finds the required number of rules with the given minimum confidence. The traditional approach can be improved by overriding some trade-off phases and discarding the unwanted objects and fields from the association analysis. The Apriori algorithm needs deep analysis, review as well as revision in terms of the inefficiencies or trade-offs for assorted applications.

The proposed approach uses reverse Apriori algorithm i.e. backtracking a database in order to find maximal frequent pattern and deriving corresponding association rules. In contrary to Apriori, Reverse approach begins with the highest occurrences of collective attributes from a database. These collective attributes are tested against the minimum support for the associated rule and is thereby selected for the next level or pruned from the subsequent levels.

5. IMPLEMENTATION SCENARIOS

Comparison of APRIORI and REVERSE-APRIORI in generation of frequent itemsets

To compare Apriori and Reverse-Apriori in generation of frequent itemsets we are considering the following implementation scenario on data fetched from web server log file. The Dataset is the transactional data fetched from a Live Web Server Log File. Consider the following database taking the support as 25%. By which each item should appear more than 25% in dataset.

5.1 Apriori

The Transactional Data from a web server log file is fetched and listed in Table 5.1 according to the web server attributes.

Table 5.1: Web Server Transactional Data

	Mailserver	HttpService	ThirdpartyAPI
		HttpService	PageNotFound
		HttpService	FTPService
	Mailserver	HttpService	PageNotFound
		Mailserver	FTPService
		HttpService	FTPService
		Mailserver	FTPService
Mailserver	HttpService	FTPService	ThirdpartyAPI
	Mailserver	HttpService	FTPService

Finding Frequent Itemsets using Apriori algorithm

To find frequent Itemsets, candidate generation is performed with 1-Itemset occurrences.

Table 5.2: 1-Itemset Occurrences

1-Itemset	Support
Mailserver	6
HttpService	7
FTPService	6
ThirdpartyAPI	2
PageNotFound	2

Since every Itemset satisfies the minimum support level, therefore none of the Itemsets will be pruned from the above candidate database. To fetch frequent patterns the 2-Itemset Combinations are generated next. The combinations are listed in Table 5.3.

Table 5.3: 2-Itemset Occurrences

2-Itemset	Support
Mailserver, HttpService	4
Mailserver, ThirdpartyAPI	2
Mailserver, FTPService	4
Mailserver, PageNotFound	1
HttpService, FTPService	4
HttpService, ThirdpartyAPI	2
HttpService, PageNotFound	2
FTPService, ThirdpartyAPI	1
FTPService, PageNotFound	0
PageNotFound, ThirdpartyAPI	0

The above table 5.3 consists of some combinations that do not satisfy minimum support level, therefore those Itemset – Combinations are pruned. The pruned database is listed in Table 5.4.

Table 5.4: Pruned Database of 2-Itemset Combinations

2-Itemset	Support
Mailserver, HttpService	4
Mailserver, ThirdpartyAPI	2
Mailserver, FTPService	4
HttpService, FTPService	4
HttpService, ThirdpartyAPI	2
HttpService, PageNotFound	2

To fetch frequent patterns from the 2-Itemset Combinations that satisfy a minimum support combinations of 3-Itemset are generated next. The combinations are listed in Table 5.5 and are checked for minimum support level for further pruning of dataset.

Table 5.5: 3-Itemset Occurrences

3-Itemset	Support
Mailserver, HttpService, ThirdpartyAPI	2
Mailserver, HttpService, FTPService	2
Mailserver, HttpService, PageNotFound	1

The above table 5.5 consists of some combinations that do not satisfy minimum support level, therefore those Itemset – Combinations are pruned. The pruned database is listed in Table 5.6.

Table: 5.6 Pruned Database of 3-Itemset Combinations

3-Itemset	Support
Mailserver, HttpService, ThirdpartyAPI	2
Mailserver, HttpService, FTPService	2

As there are no further combinations possible from Itemsets in Table 5.6, the candidate generation for 4-Itemset will not take place. Apriori terminates at this stage.

5.2 Reverse – Apriori

To find association rules form the same database using Reverse-Apriori Transactional Database listed in Table 5.1.

Finding Frequent Itemsets using Reverse-Apriori algorithm

To find frequent Itemsets, in Reverse Apriori the candidate generation is performed with 4-Itemset occurrences; the 5-Itemset generation combinations are found and listed in Table 5.7.

Table 5.7: 5-Itemset combinations

5-Itemset	Support
Mailserver, HttpService, FTPService, ThirdpartyAPI, PageNotFound	**0**

Since the 5-Itemset combinations generated does not satisfy the minimum support as listed in Table 5.7. The 4-Itemset combinations have to be generated.

Table 5.8: 4-Itemset combinations

4-Itemset	Support
Mailserver, HttpService, FTPService, ThirdpartyAPI	1
Mailserver, HttpService, FTPService, PageNotFound	0
Mailserver, HttpService, ThirdpartyAPI, PageNotFound	0
Mailserver, FTPService, ThirdpartyAPI, PageNotFound	0
HttpService, FTPService, ThirdpartyAPI, PageNotFound	0

As in the above Table 5.8 none of the 4-Itemset combinations satisfy the minimum support so all the combinations will be pruned. The 3-Itemset combinations have to be generated.

Table 5.9: 3-Itemset Combinations

3-Itemset	Support
Mailserver, HttpService, FTPService	2
Mailserver, HttpService, PageNotFound	1
Mailserver, HttpService, ThirdpartyAPI	2
FTPService, ThirdpartyAPI, PageNotFound	0
HttpService, FTPService, ThirdpartyAPI	1
HttpService, FTPService, PageNotFound	0

There are some combinations in Table 5.9 which satisfy minimum support. All those itemsets which does not satisfy minimum support are pruned and are listed 5.10.

Table 5.10: Pruned Dataset

3-Itemset	Support
Mailserver, HttpService, FTPService	2
Mailserver, HttpService, ThirdpartyAPI	2

6. RESULTS

Applying the Existing and Proposed Apriori algorithm on the live data fetched from honeypot server. After executing classical algorithm on the data to calculate 1-Itemset, 2-Itemset and 3-Itemset combinations and observing the executing time to fetch the association rules.

Proposed Approach

The above two screenshots show the execution time variation in existing and proposed approach. The proposed algorithm takes half of the execution time taken by the existing algorithm.

The graph of applying existing algorithm number of times and their execution times are plotted against the number of simulation attempts.

After 10 simulation attempts on existing Apriori algorithm is listed below:

After same 10 simulation attempts on proposed Apriori algorithm the graph is listed below:

7. CONCLUSION AND FUTURE WORK

Web usage mining is the application of data mining techniques to discover usage pattern from Web data, in order to understand and better serve the needs of Web-based applications. Web mining process can be divided into two main parts. The first part includes the domain dependent processes of transforming the Web data into suitable transaction form. This includes pre-processing, transaction identification, and data

integration components. The second part includes some data mining and pattern matching techniques such as association rule and sequential patterns. In the absence of cookies or dynamically embedded session Ids in the URLs, the combination of IP address can be used as a first pass estimate of unique users. This estimate can be refined using the referrer field. One of the simple algorithms for this is Apriori algorithm. In this research work, a new technique has been devised to discover the web usage patterns of websites from the server log files with the foundation of associations rule mining and improved Apriori algorithm. Moreover, the association analysis is not restricted to the web server log files. The work can be applied in assorted applications. The effective algorithm is implemented with the improvements of classical Apriori Algorithm. For the future work, the individual patterns of each object can be analyzed on the web server log files for deep analysis of the links, platform and behaviour of the users.

8. REFERENCES

[1] Kumar, B.S. and Rukmani, K.V., 2010, "Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms" *International Journal of Advanced Networking and Applications*, Vol. 1, Issue 6, pp. 400-404.

[2] Sharma, P. and Bhartiya, R., 2011, "An efficient Algorithm for Improved Web Usage Mining" *International Journal of Computer Technology & Applications*, Vol. 3, No.2, pp. 766-769.

[3] Martinez-Romo, J. and Araujo, L., 2010, "Analyzing Information Retrieval Methods to Recover Broken Web Links", I*n Proceedings of the 32nd European Conference on Information Retrieval, ECIR 2010,* Milton Keynes, UK, pp. 26-37.

[4] Dohare, M.P.S., Arya, P. and Bajpai A., 2012, "Novel Web Usage Mining for Web Mining Techniques" *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, Issue 1, pp. 253-262.

[5] Das, R., Turkoglu, I. and Poyraz, M., 2007, "Analyzing of System Errors for increasing a web server performance by using web usage mining", *Journal of electrical & electronics engineering*, Vol. 7, No. 2, pp. 379 – 386.

[6] Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P., 1996, "The KDD Process for Extracting Useful Knowledge from Volumes of Data", *Communications of the ACM*, Vol. 39, No. 11, pp. 27-34.

[7] Das, R., Turkoglu, I. and Poyraz, M., 2007, "Analyzing of System Errors for increasing a web server performance by using web usage mining", *Journal of electrical & electronics engineering*, Vol. 7, No. 2, pp. 379 – 386.

[8] Brachman, R.J., Anand, T., 1996, "The Process of Knowledge Discovery in Databases", *Advances in Knowledge Discovery & Data Mining,* Fayyad, U.M. - Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., Eds. AAAI/MIT Press, Cambridge, Massachusetts, pp. 37-57.