A Tree based Approach for Generating Association Rules

N.K. Sharma
Assistant Professor
Engineering College Ujjain
Ujjain, M.P., India

R.C. Jain, PhD.
Director
S.A.T.I. Engineering College
Vidisha, M.P., India

ABSTRACT

The FP-tree algorithm is one of the fastest techniques for generating frequent item set for association rule mining. Extracting frequent item set and generating association rules are two major challenges in a large student admission database. The same is tried to present in this paper with the help of sample data set.

General Terms

Data mining, Knowledge Discovery, Association Rule(s)

Keywords

Frequent Item Set, A-priori, P- tree, FP-Tree, Data set

1. INTRODUCTION

Data mining has been considered as a promising field in the intersection of databases, artificial intelligence, and machine learning. Association rule mining has a wide range of applicability. Knowledge discovery from warehouses or databases is a significant area for researchers. There are many emerging fields like Online Transactions, Web Navigation, e-commerce, e-businesses and statistics where data is being collected and stored at rapid speed; this data is also being used for mining and knowledge discovery. In this modern era new tools and techniques are needed for a new generation to extract constructive information from collecting data. At some stage, making useful data with the help of new methods and techniques is the main concern of the KDD. There are various sources of data which may be in different form so the imperative task KDD process is to make data more abstract, more compact and more useful that make data into one form. Pattern discovery or pattern extraction is the core process data mining [1] [9] [10] [11].

1.1 Data mining and knowledge discovery in the real world

The main KDD application areas include Marketing, e-Business, e-Banking, Forensic Investigation, Auto Mobiles, Telecommunications and internet. Two of them have been described [2] [3].

Marketing & e-Business: In marketing & e-business, different customer groups and purchasing behavior of the customer are analyzed finding patterns such as, If a customer buys any particular item like A and also likely to buy B stuff and C stuff. Such blueprints are important to increase the business [1].

Forensic Investigation: Forensic Investigation is the application of sciences and technologies to investigate facts of interest in relation to criminal or civil law [Wiki]. Forensic investigation is also used for collecting legal evidence from the different sources related to crime place.

2. ASSOCIATION RULES

Association rule(s) is/are a way to find out close relationship(s) among large data sets from information repository. In other word association rules are if/then statements that help uncover relationships between the huge amount of data in a relational database or other information repository like warehouse [ref]. An example of an association rule would be "If a student accesses a particular web page like scheme of the course then he/she is more likely (85%) to also will access the syllabus page of the course."

Association rules are made by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships. In data mining, association rules are useful for analyzing and predicting customer behavior.

2.1 Association rules generation

An association rule represents a close relationship in the form of A→B. It means "if event A takes place then B likely occurs". Association rules are generated using two parameters one is support and another one is confidence. Association rules are generated using user-defined minimum support and minimum confidence. To generate association rules there are three steps

- Finding all frequent patterns
- Calculate support and confidence for each item or patterns
- Generate association rules from frequent patterns

3. DATA TRANSFORMATION

Data transformation is an integral part of data mining and knowledge discovery. Transforming data allows for an increased understanding of the data and discovery of new and interesting relationships between features. There are various steps/methods required for data transformation. U. Fayyad, G.P. Shapiro and P. Smyth explain the two important steps/methods i.e. data cleaning and data access in [1] [9] as follows:

Data cleaning: The MERGE-PURGE system was applied to the identification of duplicate welfare claims (Hernandez and Stolfo 1995). It was used successfully on data from the Welfare Department of the State of Washington. In other areas, a well-publicized system is IBM is ADVANCEDSCOUT, a specialized data mining system that helps National Basketball Association (NBA) coaches organize and interpret data from NBA games (U.S. News 1995). ADVANCEDSCOUT was used by several of the NBA

teams in 1996, including the Seattle Supersonics, which reached the NBA finals.

Finally, a novel and increasingly important type of discovery is one based on the use of intelligent agents to navigate through an information-rich environment. Although the idea of active triggers has long been analyzed in the database field, really successful applications of this idea appeared only with the advent of the Internet. These systems ask the user to specify a profile of interest and search for related information among a wide variety of public-do-main and proprietary sources. For example, FIREFLY is a personal music-recommendation.

Data access: Uniform and well-defined methods must be created for accessing the data and providing access paths to data that were historically difficult to get to (for example, stored of-line). Once organizations and individuals have solved the problem of how to store and access their data, the natural next step is the question, what else do we do with all the data? This is where opportunities for KDD naturally arise.

A popular approach for analysis of data warehouses is called online analytical processing (OLAP), named for a set of principles proposed by Codd (1993). OLAP tools focus on providing multidimensional data analysis, which is superior to SQL in computing summaries and breakdowns along many dimensions. OLAP tools are targeted toward simplifying and supporting interactive data analysis, but the goal of KDD tools is to automate as much of the process as possible. Thus, KDD is a step beyond what is currently supported by most standard database systems.

4. EXISTING ALGORITHMS

4.1 AIS Algorithm

AIS (Agrawal, Imielinski, and Swami) algorithm was the first algorithm suggested for mining association rule in [5]. It pays attention on enhancing the worth of databases together with needed functionality to process decision support queries. The AIS Algorithm candidate item sets are generated and counted on the way as the database is read by the program. After reading a transaction, it is finding out which of the item sets that were found to be larger in the previous pass are contained in this transaction. New candidate item sets are generated by extending these large items sets with other items in the transaction .In this algorithm only one item consequent association rules are generated. "The main drawback of the AIS algorithm is too many candidate item sets that finally turned out to be small are generated, which requires more space and wastes much effort that turned out to be useless". At the same time this algorithm requires too many passes over the whole database [5] [6].

4.2 SETM Algorithm

The SETM algorithm was inspired by the desire to use SQL to compute large item sets [7]. The SETM algorithm also produces candidates on the-way based on transactions read from the database. It generates and counts each candidate item set that the AIS algorithm generates. It stores a copy of the candidate item set mutually with the TID of the generating transaction in a sequential structure. SETM retains information of the TIDs of the generating transactions with the candidate item sets. The main drawback of SETM is mainly due to the size of candidate sets. For each candidate Item set, the candidate set now has as many entries as the number of transactions in which the candidate Item set is present [4].

4.3 A-priori Algorithm

A-priori is a great well known approach for association rule mining was first proposed by Agrawal in 1994. The existing AIS is now a simple approach that needs numerous passes over the database, generating many candidate item sets and storing counters of each candidate while most of them make to be not frequent. Apriori is more proficient during the candidate generation process for these reasons:

a) Apriori employs a special candidate generation technique and a new pruning technique. In the process of finding frequent item sets.

b) Apriori avoids the effort wastage of counting the candidate item sets that are known to be infrequent. Due to this the number of left over candidate item sets ready for further support checking that way it becomes lesser, which considerably decreases storage requirements [6] [7].

There are few weaknesses of a-priori algorithm which makes the task tedious.

- If item set is bigger than a lot of frequent patterns are generated. It takes more time to generate frequent patterns.
- Long patterns are generated if the support threshold is low or minimum.
- Repeated database scans costly

4.4 FP-Tree

FP-Tree (Frequent Pattern Tree) Algorithm was evolved as a solution to overcome the disadvantages of Apriori algorithm. FP-Tree, frequent pattern mining, is another advance technique in the development of association rule mining, which reduces the two restrictions of the Apriori algorithm. The frequent item sets are generated with only two passes over the database and without any candidate generation process. FP-Tree was introduced by Han et. al in 2000. The omission of the candidate generation process and few passes over the database, FP-Tree is faster than the Apriori algorithm. The process of generating patterns consists two sub processes: constructing the FT-Tree, and generating frequent patterns [8].

4.5 PARTITION

The PARTITION algorithm tries to deal with two major limitations of formerly specified algorithms. The first issue with the previous algorithms is that the number of passes over the database is not known in advance. Apriori-TID tries to avoid this issue by buffering the database, but then the data base size is limited by the size of main memory. The second problem lies with pruning the database in afterward passes, i.e. removing unnecessary parts of the data. AIS and Apriori are not well fitted for this problem. This issue has been described in PARTITION algorithm.

5. PROPOSED METHODOLOGY

For the approaches specified, we adopted FP-tree algorithm which is derived from P-tree generation algorithm, which is specified below, and thereafter illustrated approach is tried to implement with an example dataset in following section.

Algorithm PTREE( )

Input: A transaction database DB and a minimum support threshold.

Output: A pattern tree

Step1: Construct a P-tree P and obtain the item frequency list L

(1) P ← Root

(2) L ← *Empty*

(3) For each transaction T in the transaction database

i) Sort T into T_i in alphabetic order. Here in each sorted transaction T = T_i, such that t is the first item of the transaction and T_i is the remaining item in the transaction.

ii) **INSERT *(T_i, P)***

iii) Update L with item in T_i

INSERT *(T_i, P)*

BEGIN

FOR each of P's child node N

IF (t.Name = N.Name)

THEN

N.freq ← N.freq+1

IF (T_i is not empty)

THEN Insert (T_i, N)

ENDIF

RETURN

ENDIF

N'.Name ← t.Name *Create a new N'*

N'.freq ← 1

P.childList ← N'

IF (T_i is not empty)

THEN **INSERT (T_i, N')**

ENDIF

RETURN

END \\ *end of function insert*

Step 2: Restructure the initial P-tree P

(1) P_{new} ← Root

(2) For each path from the root to a leaf in the initial P-tree

repeat

i) The common support of each item in P_i is that of the node next to the last branching node. If there is no branching-node in P_i, the common support of each item is the actual support of each item in P_i.

ii) Get a sub-path P_i' from P_i with the common support for every item

iii) Sort P_i' according to L

iv) Insert the sorted P_i' into the new P-tree, by calling function (P_i', P_{new})

$$P_i \leftarrow P_i - P_i'$$

untill **(P_i = Φ)**

Now, below specified is an algorithm to generate FP-Tree from obtaining P-Tree.

ALGORITHM FPTREE ( )

Input**:** A P Tree P, Item Frequency List L, and the minimum support threshold α

Output: An FP Tree

1. Frequent Item List FL ← Φ

2. For each item *i* in L

a. IF *(i.freq >= α)*

Add *i* to the FL.

3. Sort FL by frequency in descending order.

4. Invoke **CHECK (P)**.

CHECK (N)

BEGIN

FOR (each child *C* of the node *N)*

IF *C* ϵ FL

THEN

CHECK(*C*)

ELSE

Delete *C (along with sub-tree starting from C)*

END IF

END FOR

RETURN

END

Then, after performing illustrated algorithm to the available data set, appropriate rule is applied to extract knowledge required. The whole proposed methodology can be visualized by the following figure representing sequential procedure for the same.

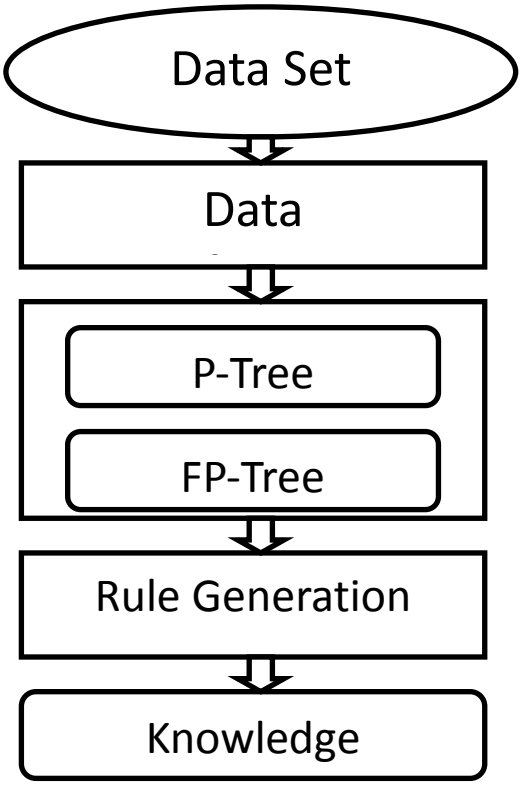

Figure 1: Proposed Approach (Pictorial view)

6. IMPLEMENTATION EXAMPLE DATA SET

To expound the effectiveness of the approach specified above a Student counseling data set has been considered(SCDS), with three attributes, namely RollNo, Student_Choice and Branch_ID. Similarly Brnch_ID consists of branch ID and college name. Following Tables show Student Admission Data Set (SADS). Relational view of the same is being illustrated below.

Table 1: Student Choice Data

Student Choice Data		
Roll No	**Student Choice_ Sequence**	**Branch_ ID**
1001	1	4
1002	2	2
1003	3	6
1004	4	11
1005	5	7
1006	6	9

Table 2: Branch ID Generation

Branch ID Generation		
Branch _ ID	**COLLEGE_NAME**	**BRANCH**
1	C1	CS
2	C1	IT
3	C1	EC
4	C2	CS
5	C2	IT

Above specified data is then treated with the said approach and transactional data set is obtained and same for above given data is obtained and expounded below.

Table 3: Transactional Data Set

Transactional Data Set				
Transaction_ID	**Item1**	**Item2**	**Item3**	**Item4**
T1	1	1	1	1
T2	1	0	1	1
T3	1	1	0	1
T4	1	0	1	1
T5	1	0	1	1
T6	1	0	1	1

7. RULES GENERATED AND RESULT ANALYSIS

7.1 Rules Generated

Rules are generated by taking minimum support is 75% and minimum confidence is 80%. The generated rules are shown in Table 4 below:

Table 4: Rules Generated

Rule	Confidence
C1 -> C3	83.33333
C3 -> C1	100
C1 -> C4	100
C4 -> C1	100
C3 -> C4	100
C4 -> C3	83.33333
C1, C3 -> C4	100
C4 -> C1, C3	83.33333
C1 -> C3 ,C4	83.33333
C3, C4 -> C1	100

7.2 Result Analysis

Depending on the number of transactions the run time is calculated using Apriori algorithm as well as our approach and the comparison is expounded below:

Table 5: Result Analysis

S. No.	No. of Transactions	Run Time using Apriori (in seconds)	Run Time using Our approach (in seconds)
1	1000	7	5
2	5000	15	9
3	7500	28	16
4	10000	45	23

8. CONCLUSION

For mining association rule there founds a common drawback of various scans over the database in every algorithm, this drawback can be overcome by introducing data transformation approach. FP-tree algorithm which is derived by P-tree generation algorithm has been used to extract interesting patterns and to develop significant relationships among variables stored in a huge dataset.

The mined association rules reveal various factors for admission in professional courses like student's college and branch choices, place, past placement status, available infrastructure etc. The approach is based on real time data used for admission in various engineering institutions; the same can be improved when approached with other techniques.

9. ACKNOWLEDGEMENTS

Our sincere thanks to our college **Mr. Manoj Yadav**, Software Consultant in Bhopal, Madhya Pradesh, India for the immense support you have provided us for publishing this paper.

10. REFERENCES

[1] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth, "*From Data Mining to Knowledge Discovery in Databases*", AAAI, 1996.

[2] Andreas Mueller, "*Fast Sequential and Parallel Algorithms for Association Rule Mining: A Comparison*", 1995.

[3] Agrawal, R., and Psaila, G. 1995. Active Data Mining. In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95). Menlo Park, Calif.: American Association for Artificial Intelligence.

[4] M-S Chen, J Han and P. S. Yu, Data Mining : An Overview from a Database Perspective, IEEE Tran. On Knowledge and Data Engg., December,1996.

[5] R. Agrawal, T. Imienski and A. Swamy, Database Mining : A Performance Perspective, IEEE Tran. On Knowledge and Data Engg., December,1991.

[6] Rakesh Agrawal, and Ramakrishnan Srikant, "*Fast Algorithms for Mining Association Rules",* Proceedings of the 20th VLDB Conference Santiago,Chile,1994.

[7] M.Houtsma and A.Swami. Set-oriented Mining of Association Rules. Research Report RJ 9567, IBM Almaden Research Center,SanJose,California,October1993.

[8] Qiankun Zhao and SouravS. Bhowmick, "Association Rule Mining: A Survey", Singapore.

[9] U. Fayyad, G.P. Shapiro and P. Smyth, The KDD Process for extracting Useful Knowledge from Volumes of Data, Communication of the ACM, Nov., 1996.

[10] T. Imielinski and H. Mannila, A Database Perspective on Knowledge Discovery, Communication of the ACM, Nov., 1996.

[11] A. Sawasere, E. Omiecinski and S. Nawathe, An Efficient Algorithm For Mining Association Rules In Large Databases, Proceedings Of The 21st VLDB Conference, Zurich, 1995.