Scientific Bulletin of the „Petru Maior" University of Tîrgu Mureş
Vol. 12 (XXIX) no. 1, 2015
ISSN-L 1841-9267 (Print), ISSN 2285-438X (Online), ISSN 2286-3184 (CD-ROM)

STEP-BY-STEP MODEL FOR THE STUDY OF THE APRIORI ALGORITHM FOR PREDICTIVE ANALYSIS

Daniel Grigore ROŞCA[1], Dumitru RĂDOIU[2]
[1,2] "*Petru Maior*" *University of Tîrgu Mureş*
Nicolae Iorga Street, no. 1, 540088 Tîrgu Mureş, Romania
[1]rosca.daniel.grigore@stud.upm.ro
[2]dumitru.radoiu@science.upm.ro

Abstract

The goal of this paper was to develop an educational oriented application based on the Data Mining Apriori Algorithm which facilitates both the research and the study of data mining by graduate students. The application could be used to discover interesting patterns in the corpus of data and to measure the impact on the speed of execution as a function of problem constraints (value of support and confidence variables) or size of the transactional data-base. The paper presents a brief overview of the Apriori Algorithm, aspects about the implementation of the algorithm using a step-by-step process, a discussion of the education-oriented user interface and the process of data mining of a test transactional data base. The impact of some constraints on the speed of the algorithm is also experimentally measured without a systematic review of different approaches to increase execution speed. Possible applications of the implementation, as well as its limits, are briefly reviewed.

Key words: data mining, apriori,business intelligence, actionable rule

1. Introduction

Data mining is a business intelligence process based on extracting new data (e.g. hidden patterns) from available data sets [4]. To perform data mining we need a mining algorithm able to identify patterns (e.g. association rules) which could be used in forecasting.

Today almost all companies collect data and try to extract from it new information, process known as business intelligence. Some of the most advanced users of data mining technology are credit and insurance companies with the goal to identify singularities and patterns which could be linked to fraud or decision making. Others, e.g. car manufactures, collect data to predict when different parts of a machine will fail and forecast necessary spares. Retailers use data mining (market basket analysis) to identify association rules for insight into which products will be purchased together (e.g. A is frequently bought with B). The identification of a pattern which supports a future action (actionable rule) is then used as basis for marketing decisions. E.g. retailers could offer a discount only to one item (A) as the other (B) will be almost certainly bought at full price or higher than competitors' price.

2. The Apriori Algorithm

Apriori is a classical algorithm used in data mining for the discovery of the association rules in transactional data. The algorithm is designed to work with huge amounts of data representing sequences of items or events. Examples: a collection of events generated by visitors of an on-line store or a collection of sequences of items (representing sale transactions) purchased in a supermarket.

In transactions, data can be viewed as a sequence of items from a collection/set, where each item from a sequence, has an associated (most of the time unknown) reason for being in that transaction. E.g. items bought by a customers in a supermarket.[9]

In this example, all items sold by the supermarket form an item set, I

$$I = \{A,B,C,\ldots Z\}$$

A sequence of items bought together by a customer forms a transaction. E.g. transactions: [A,D] [A,B] [B,D,A].

If the sequence of items is strict, the transaction are called serial (e.g. DNA sequencing). In our example the order is not strict; the transactions are called parallel.[8]

The Apriori algorithm is used on collections of parallel transactions, called in the paper transaction data bases.[6]

A transaction may appear in the database only once or several times. Using again the supermarket example, is easy to see that transactions which appear more frequent are of interest because they signal a behavioral pattern of the customer (which pattern should be identified and used for further decisions)[7]

A transaction is considered frequent if its frequency is above some user defined frequency threshold. E.g. In (Table 1), threshold frequency value is 2 and transaction [D,E] is not frequent.

Transaction	Frequency
A,D	5
A,B	3
B,D,A	3
D,E	2

Table 1: Frequent transactions in a database

The goal of the Apriori algorithm is generate the association rules for items from frequent transactions, i.e. Rule: The person acquiring item A is very likely to acquire item B or D.

An association rule is an implication expression of the form [5]:

$A \Rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \emptyset$

The process of mining for association rules is about discovering a set of rules that is shared among a large percentage of the data [3].

In ordering frequent transactions, a reference numerical value could be used: support counting.

Support is the ratio of the number of transaction of the type A,B and the total number of transactions in the corpus of data.

Support [A,B] = n/N

Where n is the number of frequent transactions of the type [A,B] and N is the total number of frequent transactions.

According to Apriori Principle [1], if a transaction is frequent, all of its subsets must also be frequent.

Confidence is introduced to numerically quantify the relevance of an item in a transaction containing several items.

Confidence is the ratio of the number of transactions of the type A,B and the total number of transactions containing item A.

Confidence [A,B] = n/N[*,*,A] where N[*,*,A] is the total number of transactions in which item A appears and n is the number of transactions of the type [A,B].

3. How the algorithm works

Step 1:

Choose a frequency threshold so that the corpus of data contains only what you consider frequent transactions.

Generate all possible 1-transactions from an item set [A],[B],[C],[D],…,[Z]

Compute the frequency (how many times the transaction appears in the data base) of each 1-transactions.

Compare the frequency of all the 1-transactions with the frequency threshold. If the frequency is smaller than the frequency threshold the 1-transaction is removed.

Step 2:

Join the transactions to generate a set of candidate k-transactions([A,B], [B,C]…) And use Apriori property (frequency < frequency threshold) to eliminate the unfrequently k-transactions.

Step 3:

Scan the transaction database to get the frequency of candidate k-transactions and compare with the frequency threshold and get a set of frequent k-transactions.

Step 4:

Repeat step 2 until the set of candidate k-transactions list is empty.

If the set of candidate k-transactions list is not empty move to step 5.

Step 5:

For each frequent transaction F ([A,B,C]), generate all nonempty transactions of F ([A],[B],[C],[A,B],[A,C],[B,C]).

Step 6:

For every nonempty transactions of F, output the rule "[A]=>(F-[A])" if confidence C of the rule "[A]=>(F-[A])" = frequency of F / frequency of [A]) >= minimum confidence[10]

4. The Educational Tool

Fig. 1: GUI of the educational tool

Fig. 2: GUI for adding a transaction

To compute the frequent transactions and to generate association rules, the data base has to be loaded using a CSV file containing all the transactions.

4.1 Preprocessing

Transaction data base have to be read from a CSV file with a table contains 1 column with the TID (transaction ID – unic number) and n-columns with transactions.

The threshold frequency has to be set before running the Apriori Algorithm.

4.2. User Interface

Main Window (Fig. 1)

Window no. 1 contains a formatted view of the transactions data base.

The first column in the transaction table contains the transactions ID; the second column contains the items of each transaction.

Window no. 2 contains the list of the candidate k-transactions with the frequency computed.

The table contains the list of all candidates: first column represent transactions, second column represent the frequency of the transaction in the data base.

This table will help students to identify all transactions which do not fulfil the Ariori condition (frequency < frequency threshold).

Window no. 3 contains the list with the frequent transactions after the use of the Apriori property (frequency < frequency threshold) to eliminate the unfrequented k-transactions.

Window no. 3 contains the table with the transactions fulfilling the Apriori condition.

Window no. 4 contains the content of the CSV file.

Area no. 5 contains the buttons to add, to remove transactions. Also it contains the buttons for uploading the CSV file with the transactions data base

The controls in this area allow students to add or delete transactions.

Area no. 6 contains the support text that explains step by step the process of the algorithm.

4.3 Adding a transaction window (Fig. 2)

Window no. 7 contains the list of predefined items for creating transactions.

The student can select any combination of items to create a transaction.

Window no. 8 contains the selected items for the transactions.

The student can add the generated transaction.

5. Testing if the Apriori Algorithm depends on the support counting

Choosing higher support and confidence values and reducing the set of frequent transactions. In the paper this was the only approach tested to see if there is an impact on the execution speed. The teste results presented bellow suggest (without a systematic analysis) that higher values for support and confidence values have limited influence on the execution speed therefore another method must be identified to speed up algorithm execution.

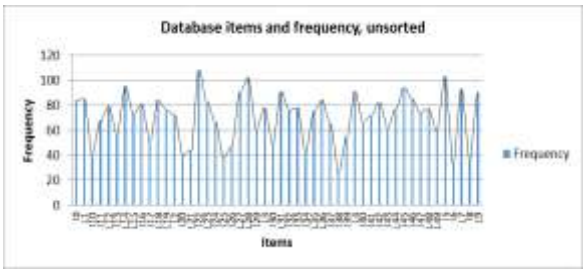

Fig. 3: Database visualized as items and frequency (unsorted)

Fig. 4: Database visualized as items and frequency (sorted)

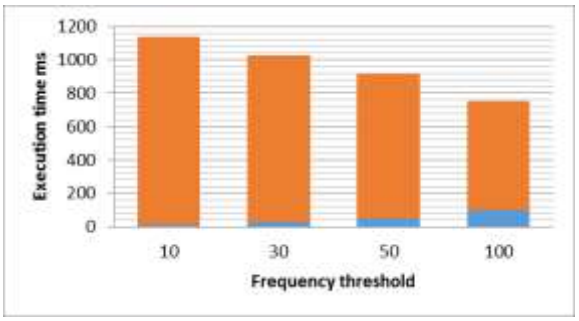

Fig. 5: Execution speed variation with frequency threshold

Threshold	Execution time ms
10	1132
30	998
50	866
100	654

Tabel 2: Execution speed for different threshold values

(Fig. 3) represent transactions in the data base and their frequency. The algorithm does not sort the elements in a transaction. (Fig. 4) represent transactions in the data base and their frequency after the sorting process. (Fig. 5) represents the dependence of execution time with the frequency threshold. The chose frequency threshold are: 10, 30, 50, 100 (Table 2) on a set of 1000 transactions. We can notice that the higher the frequency threshold, the lower the execution time. For instance, frequency threshold increased ten times (from 10 to 100) leads to a 40% decrease of execution time.

6. Conclusions

The goal of this paper was to develop an educational oriented application based on the Data Mining Apriori Algorithm which facilitates both the research and the study of data mining by graduate students. The Apriori algorithm has both advantages and disadvantages (). It is easy to implement, simple to use and relatively easy to parallelize for huge datasets. The major disadvantage is speed. The algorithm requires a large number of data base scans over data which must be resident in the transactional database.

Apriori algorithm can be very slow and the bottleneck is candidate generation.

Several methods have been identified to improve algorithm efficiency:

- Sampling very large datasets (mining a subset of the given data)
- Dynamic counting (add new candidate only if all their subsets are estimated to be frequent
- And choosing higher values for support and confidence

We plan to analyze in a systematic manner which of the methods is the most efficient and to implement a function which supports this in the next version of the algorithm implementation.

References

[1] Tan, Pang-Ning, Steinbach, Michael, Kumar, Vipin (2006), Introduction to data mining, Pearson Addison Wesley Boston

[2] http://www.slideshare.net/INSOFE/apriori-algorithm-36054672, Retrieved 01.05.2015

[3] Zaki, M. J. (2000), Scalable Algorithms for Association Mining, IEEE Transaction on Knowledge and Data Engineering, Vol. 12, No. 3, pp. 372-390.

[4] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. (1996), From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press, pp. 1-34.

[5] Agrawal, R., Imielinski, T., Swami, A. (May 1993), Mining association rules between set of items in large databases, Proceedings of ACM SIGMOD, pp. 207-216.

[6] Jiawei, Han and Micheline, Kamber (4 Jun 2006), Data Mining: Concepts and Techniques, 2 edition.

[7] Gordon, S. Linoff, Michael, J. Berry (1 April 2011), Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management, 3rd Edition

[8] Seng, J., Chen, T. C. (2010), An analytic approach to select data mining for business decision. Expert Systems with Applications, 37, pp. 42-57.

[9] Tsai, P. S. M., Chen, C. (2004), Mining interesting association rules from customer databases and transaction databases. Information Systems, 29, pp. 685-696.

[10] Liao, S., Chen, C., Wu, C. (2008), Mining customer knowledge for product line and brand extension in retailing. Expert Systems with Applications, 34, pp. 63-76.