Association Rule Mining and Website's Design Improvement

Asem Omari and Stefan Conrad
Heinrich-Heine University Düsseldorf
Institute of Computer Science
Databases and Information systems
Düsseldorf, Germany
{omari, conrad}@cs.uni-duesseldorf.de

Abstract

Many commercial companies collect large quantities of data from daily operations. For example, customer orders or purchase data are collected daily at the counters of grocery stores. Data mining is applied to such kind of data to extract patterns that could be useful to learn about the purchasing behavior of the customers. Such information can be useful to support a variety of business related tasks. For example, the investment of that kind of information in building the company's website. Association rule mining is one of the techniques used to mine databases. Association rule mining is the discovery of association rules showing attribute values that occur frequently together. In this paper, we discuss different association rule mining techniques and algorithms. Then, we will find out which techniques and algorithms could be best used to produce useful patterns. Finally, we will see how to invest that patterns to improve the structure of the company's website in its design phase.

1 Introduction

Data Mining is the extraction of useful patterns from large databases. Researchers attract much attention to Data Mining because of its wide applicability in many different fields. One major application area of Data Mining is mining Association Rules among items in a database of sales transactions which is known as Market Basket Analysis. The rules extracted using such a Data Mining technique can be used to support a variety of business related tasks. In our approach, we want to get benefit from extracted patterns to support the design of the company's website that the transactions database belongs to. This is done in the design phase through improving the structure of the website depending on the extracted patterns in a way that makes it easy for the website's navigator to find his target products in an efficient time, give him the opportunity to have a look at some products that may be of interest for him, and encourage him to buy more from the available products which will consequently increase the company's overall profit. This paper is structured as follows: In section 2, we give an overview about Association Rule Mining. In section 3, we will discuss different common Association Rule Mining techniques and algorithms, and make comparisons between them with respect to some attributes. Then, we will decide which techniques or algorithms can be beneficial to our approach. Then, in section 4, we will see how we can use the extracted patterns to improve website's design structure. Finally, in section 5, we summarize our paper and present out future work.

2 Association Rule Mining

In our previous work [1] we discussed the usage of Data Mining to support website's designers to have better designed websites. Different Data Mining techniques can be used to support website's design. Association Rule Mining is one of the Data Mining techniques that plays an important rule in our approach. An Association Rule is an expression of the form $X \Rightarrow$Y, where X and Y are sets of items and have no items in common. Given a database of transactions D where each transaction $T \in$D is a set of items, $X \Rightarrow$Y denotes that whenever a transaction T contains X then there is a probability that it contains Y too. The rule $X \Rightarrow$Y holds in the transactions set T with confidence c if a determined percentage of transactions in T that contain X also contain Y. The rule has support s in T if a determined percentage of all transactions in T contains both X and Y. Association Rule Mining is finding all Association Rules of which the support is greater than or equal a user-specified minimum support (*minsup*), and minimum confidence *(minconf)*. In general, the process of extracting interesting Association Rules consists of two major steps. The first step is finding all itemsets that satisfy minimum support (known as *Frequent-Itemset* generation). The second step is generating all association rules that satisfy minimum confidence using itemsets generated in the first step. In this paper we concentrate our discussion on the step of finding frequent itemsets.

3 Common Association Rule Mining Algorithms

According to [7], in the process of searching for frequent itemsets, Association Rule Mining algorithms employ one of two common approaches: Breadth-first search Approach (BFS), and Depth-first search approach (DFS). In BFS approach, the support values of all $(k-1)$-itemsets are determined before counting the support values of the k-itemsets where k is a positive integer. Supposing that the transactions data is represented in a tree structure, in DFS approach the algorithm can start from, say, node a in the tree and counts its support to determine whether it is frequent. If so, the algorithm expands the next level of nodes until an infrequent node is reached. It then backtracks to another branch and continues the search from there. In the subsections below we briefly describe the basic principles and differences between common Association Rule Mining algorithms.

3.1 The Apriori Algorithm

The Apriori algorithm [4] follows the Breadth-first search strategy. It generates all frequent itemsets, called also large itemsets, by making multiple passes over the transactions database D. The algorithm makes a single pass over the data to determine the support of each item which results in the set of 1-itemsets. Next, the algorithm will iteratively generate new candidate k-itemsets using the frequent $(k-1)$-itemsets found in the previous iteration. An additional pass over the data set is made to count the support of the candidates. The algorithm eliminates some of the candidate k-itemsets using the support-based pruning strategy. If any subset of the k-itemset X is not frequent then X is pruned. After counting their supports, the algorithm eliminates all candidate itemsets whose support count are less than *minsup*. The algorithm terminates when there is no new frequent itemset generated. Association rules are generated by generating all non-empty subsets of each frequent itemset and outputs the rule if its confidence is $\geq$ *minconf*.

3.2 The AprioriTID Algorithm

The AprioriTID algorithm differs from the Apriori algorithm in that it does not use the database D for counting support after the first pass. Rather, it uses the set of candidate k-itemsets associated with the transactions identifiers (TID's). So that the number of entries in this set may be smaller than the number of transactions in the database, especially for large values of k [8].

3.3 The AprioriHyprid Algorithm

The AprioriHyprid algorithm is an algorithm that get benefit from both Apriori and AprioriTID algorithms. It starts by using the Apriori algorithm, then it switches to AprioriTID in the last passes. But if there are no candidate itemsets found in this stage then we just pay the cost of switching to AprioriTID without getting benefit of using it [8].

3.4 FP-Growth Algorithm

Unlike the Apriori algorithm, the FP-Growth algorithm follows the Depth-first search strategy. FP-Growth mines frequent itemsets without candidate generation. It encodes the data set using a compact data structure called the **FP-Tree**, and extracts frequent itemsets directly from this structure. The FP-tree is created as follows [2]: First, a root node of the tree is created and labelled "null". For each transaction in the database, the items are processed in reverse order and a branch is created for each transaction. Every node in the FP-tree stores a counter which keeps track of the number of transactions that share that node. When adding a branch for each transaction, the count of each node among the common prefix is incremented by 1, and nodes for the items in the transaction following the prefix are created and linked accordingly. Additionally, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links. Each item in this header table also stores its support. The transactions in the FP-tree are stored in support descending order which keeps the FP-tree representation of the database as small as possible since the more frequently occurring items are arranged closer to the root of the FP-tree and thus are more likely to be shared. Because there are often a lot of sharings of frequent items among transactions, the size of the tree is usually much smaller than its original database which saves the costly database scans implemented by Apriori-like algorithms.
The study in [6] shows that FP-Growth has several advantages than any other Apriori-like algorithms, especially when the dataset contains many patterns and/or when the frequent patterns are long. In an experimental application on different real-world and artificial datasets in [9], a comparison between Apriori, FP-Growth and other algorithms shows that Apriori outperforms FP-Growth when the *minsup* vlaue is small. But with high *minsup* values, FP-Growth outperforms Apriori. Another experiments on real-world and artificial datasets are done by [3]. The experiments show that FP-Growth performs best in comparison with Apriori and other implemented algorithms. The FP-Growth mining method is implemented in the *DBMiner* system [5].

4 Supporting Website's Design Using Association Rule Mining

After generating frequent items, association rules that are $\geq minconf$ are generated. These rules are called interesting association rules. These rules can be invested in many different applications. One of these applications is improving the structure of the company's website that the mined database belongs to. This is done in the website's design phase by creating

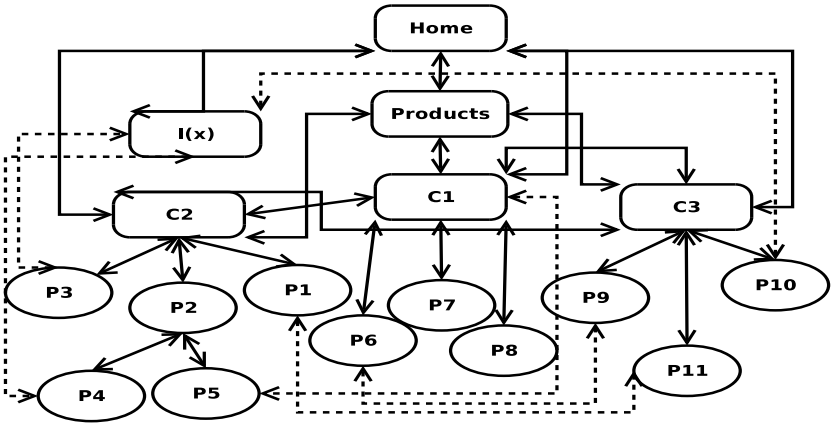

Figure 1: Improved Website's Design Structure Using Extracted Association Rules

links between items that seem to be sold together, or highlight that links if they already exist, and/or create index pages which are pages that have direct links to some products that may be of interest for some group of customers.

Figure 1 represents a part of the website's structure of a company that sells different kinds of products. Boxes and circles represent pages of the website, and arrows represent links between pages. *C1*, *C2*, and *C3* represent different product categories, and *P1*, ..., *P11* represent different products belonging to these categories. The dotted arrows represent links created with the help of the extracted interesting association rules. Note that links between product-to-product, and product-to-category can be created. For example, direct links between product $P1$ and product $P11$, and product $P6$ and product $P9$ are created depending on some extracted association rules. For example, the link between product $P1$, and product $P11$ is created depending on a rule that says:

$P1 \Rightarrow P11$ [$support = 4\%$, $confidence = 60\%$]

which means that 60% of customers who buy product $P1$, buy also product $P11$, and 4% of all customers buy both products. Furthermore, a link between product $P5$ and category $C1$ is created depending on the rule that says:

$P5 \Rightarrow C1$ [$support = 5\%$, $confidence = 80\%$]

which means that 80% of customers who buy product $P5$ buy also a product belonging to $C1$ category, and 5% of all customers buy product $P5$ and one of $C1$ products. $I(X)$ is an index page that has direct links to products $P3$, $P4$, and $P10$. These products may be of interest for group X of customers. This index page may be created depending on a set of similar rules such as:

$age(X, \text{"20...35"}) \Rightarrow P3$

$age(X, \text{"20...35"}) \Rightarrow P4$

$age(X, \text{"20...35"}) \Rightarrow P10$

The previous rules mean that customers who are between 20 and 35 years of age are interested in buying products $P3, P4$ and $P10$ respectively. Consequently, such modifications done to the website's design help customers to find their target products in an efficient time, encourage them to buy more from the available products, and give them the opportunity to have a look at some products that may be of interest for them, which will consequently increase the company's overall profit.

5 Summary and Future Work

In this paper we discussed different common Association Rule Mining algorithms, and we made comparisons between them for the purpose of using them in the extraction of interesting rules from a database of product transactions. Such rules can be used to improve the design of the company's website that the transactions database belongs to. Many improvements and modifications can be done to the website's design such as, adding/modifying links, and/or creating index pages. Apriori and FP-Growth are the most used algorithms to extract association rules because they need less execution time, and less memory in comparison with other algorithms presented in literature. In most cases, the FP-Growth algorithm outperforms other algorithms including Apriori algorithm. FP-Growth needs less execution time, and less memory usage to generate frequent itemsets. As a future work, we plan to use that algorithm to generate association rules from real-world, and artificial datasets, and use the extracted association rules to support and improve the structure of the website of the company that the datasets belongs to.

References

[1] A. Omari, and S. Conrad. On the Usage of Data Mining to Support Website Designers to Have Better Designed Websites. In P. Dini, P. Lorenz, D. Roman and M. Freire, editor, *International Conference on Internet and Web Applications and Services (ICIW06)*, Guadeloupe, French Caribbean, 19-25 February 2006. IEEE Computer Society.

[2] B. Goethals. Survey on Frequent Pattern Mining: A Manuscript. http://www.adrem.ua.ac.be/bibrem/pubs/fpm_survey.pdf , 2003.

[3] C. Borgelt. An Implementation of the FP-growth Algorithm. In *Workshop Open Source Data Mining Software (OSDM'05, Chicago, IL)*, pages 1–5. ACM Press, 2005.

[4] J. Han and M. Kamber. *Data Mining Concepts and Techniques.* Morgan Kaufmann Publishers, San Francisco, 2001.

[5] J. Han, J. Chiang, S. Chee, J. Chen, Q. Chen, S. Cheng, W. Gong, M. Kamber, K. Koperski, G. Liu, Y. Lu, N. Stefanovic, L. Winstone, B. Xia, O. R. Zaïane, S. Zhang and H. Zhu. DBMiner: A System for Data Mining in Relational Databases and Data Warehouses. In *IBM Center of Advanced Studies Conference (CASCON)*, pages 249–260, 1997.

[6] J. Han, J. Pei, Y. Yin and R. Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[7] J. Hipp, U. Güntzer and G. Nakhaeizadeh. Algorithms for Association Rule Mining – A General Survey and Comparison. *SIGKDD Explorations*, 2(1):58–64, July 2000.

[8] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In J. B. Bocca, M. Jarke and C. Zaniolo, editor, *Proc. 20th International Conference Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 1994.

[9] Z. Zheng, R. Kohavi, and L. Mason. Real World Performance of Association Rule Algorithms. In F. Provost and R. Srikant, editor, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 401–406, 2001.