Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 62 (2012) 452 – 456

Procedia
Social and Behavioral Sciences

WCBEM 2012

A recommendation engine by using association rules

Ozgur Cakir[a] [1], Murat Efe Aras [b]

[a] *Marmara University, İstanbul, Turkey*
[b] *PortakalOfis Software, İstanbul, Turkey*

Abstract

This study represents a recommendation engine which was developed to personalize an e-commerce website. Here, the personalization approach is collaborative filtering and the technique is association rule mining. The software was developed by the programming language C# and association rules were generated by Apriori algorithm. The recommendation engine had been tested by existing data before it was deployed to an e-commerce website. Testing phase was evaluated by accuracy and coverage while the deployment phase was evaluated by basket ratio, which is the ratio of the number of products added to the shopping cart to the number of keywords searched by users. The application has taken three weeks. Results show that the recommendation engine increases the basket ratio.

Keywords: recommendation engine, association rule mining, e-commerce, basket ratio

1. Introduction

Web pages are one of the most important data repositories. Transactions and data contained in these repositories are increased every day. Because of this fact, data mining techniques are often applied on web page data. Association rule mining, which is a common data mining method, is used for extracting useful patterns among data items. These patterns refer to association rules, which are important tools to develop recommendation systems. A recommendation system can help to model web user's behavior and to predict new user's behavior. Predictions are offered to the user and it is expected that conversion rate would increase. Here, the conversion rate is the percentage of visitors who take a desired action.

In this study, we developed a recommendation engine by using association rule mining for an e-commerce website. This engine analyzes data and generates association rules based on keywords that are used for searching and products that are added to shopping cart. So the aim of this study is to offer products, which users might be highly interested, and to get higher conversion rate.

[1] Tel: +90-212-507-99-25
E-mail: ocakir@marmara.edu.tr

doi:10.1016/j.sbspro.2012.09.074

2. Literature Review

Websites, especially commercial ones, face diverse visitors who may be searching for something different. But traditional approaches that suppose these diverse users to be in one category may have some problems to react their personal needs or concerns. On the other hand, personalization techniques can help to present different contents to each user or a group of users.

Actually, there are three main different approaches to website personalization. These are content-based, rule-based and collaborative filtering systems. In content-based filtering, the content descriptions of items are represented by a user profile and items that are similar to the user profile are recommended to the user. In rule-based filtering, decision rules, often based on demographics, psychographics, or other personal characteristics of users, are used to recommend items to users. In collaborative filtering, the ratings of a current user for objects are matched with those of similar users in order to produce recommendations for objects not yet rated or seen by an active user [Mobasher].

In collaborative filtering, the system not only uses the profile for the active user but also maintains a database of other users' profiles. While traditional collaborative filtering only uses rating data, hybrid collaborative approaches that utilize both content and user rating data have also been proposed [Melville].

A recommendation engine, which is a web-based interactive software agent, can make a web page to be personalized. The task of a recommendation engine is to compute a recommendation set for the current user session, consisting of the objects that most closely match the current user profile [Mobasher et al.].

Association rule mining, which is one of the most important techniques of data mining, searches for interesting relationships among items in a given data set. Association analysis is the discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data. Association analysis is widely used for market basket or transaction data analysis [Han].

An association rule consists of two Boolean propositions and states that if the left-hand side (the antecedent) is true, then the right-hand side (the consequent) is also true. But a probabilistic rule states that the right-hand side is true with probability p, given that the left-hand side is true [Hand].

Actually, two main criteria, which are support and confidence, are used for generating association rules. Support (s) is the percentage of transactions that contain both A and B. And confidence (c) is the percentage of transactions containing A that also contain B. Another criteria is lift which gives the strength of the relationship between items A and B. A lift value greater than 1 indicates there is a positive association, whereas a value less than 1 indicates there is a negative association [Giudici].

Association rule mining is usually decomposed into two sub-problems. One is to find frequent itemsets. The second is to generate association rules from frequent itemsets with the constraints of minimal confidence. Since the second sub-problem is quite straight forward, most of the researches focus on the first sub-problem [Kotsiantis].

Helgard and Hipp reviews the most common algorithms for finding frequent items and generating association rules. One of the well-known algorithms is Apriori, which is first introduced in 1994 [Agrawal]. Main strategy of Apriori is to reduce candidate frequent itemsets. Apriori principle states that if an itemset is frequent, then all of its subsets must also be frequent.

Conversely, if an itemset is infrequent, then all of its supersets must be infrequent too. This strategy of trimming the exponential search space based on the support measure is known as support-based pruning. Apriori algorithm initially considers every item as a candidate 1-itemset. After counting their supports, the candidate itemsets that appear in fewer transactions than minimum support count are discarded. In the next iteration, the candidate 2-itemsets are generated using only the frequent 1-itemsets because of the Apriori principle. So we only need to keep candidate itemsets whose subsets are frequent [Tan].

In this study, we use Apriori.exe in order to generate association rules. Apriori.exe was developed by Christian Borgelt, who share both this file and its source codes in his website [http://www.borgelt.net/apriori.html]. Apriori.exe can report frequent itemsets, maximal frequent itemsets and closed frequent itemsets while it generates association rules.

3. Application

This study covers to construct a recommendation engine and to deploy it for an e-commerce website. The recommendation engine takes into account two main attributes. One is keyword that is searched and the other is product that is added to the shopping cart. The application has several modules that work in coordination

Actually, the process works in four main steps. At first step, log files are read and written to a text file. At second step, Apriori.exe reads the text file in order to generate association rules and writes these rules to another text file. At third step, association rules generated by Apriori.exe are read and kept in memory. And as the last step, products that are the most likely to be added to the shopping cart are displayed to the user.

3.1. Software

The application was developed by C# programming language and works on .NET framework. .NET framework supplies the communication between the application and operating system. Moreover, the application has a part that is tasked with displaying recommendations. That part is developed as an ASP.NET application. That is, the application never communicates with operating system directly. Application consist of five modules which are session module, time module, Apriori.exe, report module, pool module, display module, test module.

A session starts with a request and ends in thirty minutes after the last request by the user. Time module periodically connects web server and downloads IIS logs, which had not been analyzed yet. Session module reads those files and writes sessions to a text file. Moreover, session module identifies the user through cookies that contain username. Thereby, keyword that is searched and products that are added to the shopping cart are determined for each user.

All requests, except for searching and adding to shopping cart, are ignored since they are not used for generating association rules. So a text file, which is written by session module, contains keywords and product codes. Session module gets start Apriori.exe after the end of its task.

Apriori.exe takes into account the text file written by session module as input. It analyzes transactions in that file and generates association rules. Then, association rules generated by Apriori.exe are written to another text file, which also contains confidence and support values. Apriori.exe works on Windows command prompt.

Report module is web-based software developed by C#. It gives reports such as usage frequency of an association rule, numbers of displaying, clicking and adding to shopping cart based on rules and proportions of clicks to the number of displaying and to the number of adding to the shopping cart based on rules.

Pool module matches keywords with association rules. It is never needed to reach web server's hard disk because all existing rules had been loaded to the memory by pool module. Because of that, success is closely related to the performance of this module. Association rules are kept in a hash table where the antecedent is the keyword and the consequents are a linked list of products.

Display module has two main tasks. The first is to collect user's data such as searched keywords, clicked offers and added products to the cart. The second is to send keywords to pool module and to display the offers sent by pool module. Display module selects five products according to confidence of the rule and shows them as a small picture with name and price.

Test module records if any offered product added to the cart after a keyword was searched. Thereby, it changes the value of variables named, (1) *coverage* if any offer could be made, (2) *accurate* if any offered product had added to the cart, (3) *not accurate* if any offered product had not added to the cart and (4) *not coverage* if any offer could not be generated. These criteria are used for improving the recommendation system.

3.2. Evaluation

The software has been evaluated in two different stages. In the first stage, model has been driven on test data and model success has been evaluated by two criteria, which had been determined by test module. These criteria are accuracy and coverage. Accuracy is calculated as the proportion of the number of offers that result in a product added to the cart to the number of searches. Coverage is calculated as the number of offers to the number of searches.

In the second stage, model has been deployed an e-commerce website and commercial success has been evaluated by basket ratio, which is the proportion of the number of products added to the cart to the number of searches. This stage has taken three weeks. First week, the software was inactive and there was no any offer. Second week, software has been activated and offers has been made by association rules. Third week, the software was inactive again but offers were made randomly. The function Randomize() of C# is used for random offers.

Statistical differences between basket ratios of two weeks are tested by standard normal distribution (z) test under assumption of satisfied sample size. The null and the alternative hypothesis are given below.

$H_o: p_i = p_j \qquad H_1: p_i \neq p_j$

Here, p_i is the basket ratio for i*th* week and p_j for j*th* week. Null hypothesis is tested at significance (α) 0.05 and rejected if p-value is smaller than α.

3.3. Findings

In the first evaluation stage, the software has been run on data sets, which have different volume. A heuristic method has been carried out in order to find optimal performance in terms of coverage, accuracy and processing time. The results have showed that optimal values were obtained with 25 days data with %1 confidence. Thus, 87.74% coverage and 16.43% accuracy has been achieved in 318 minutes processing time as optimal.

Table 1. Results of deployment stage

Weeks	No. of Searches	No. of Adding to Cart	Basket Ratio (%)
1	153322	23467	15.31
2	158232	32878	20.78
3	162934	24989	15.34

The table above shows the results of deployment stages. The difference between basket ratios of Week 1 and Week 2 is statistically significant. Z-value is calculated as 39.83 with the significant 0.00 for one-tailed test. The difference between basket ratios of Week 1 and Week 3 is not statistically significant. Z-value is calculated as 0.23

with the significant 0.81 for two-tailed test. The difference between basket ratios of Week 2 and Week 3 is statistically significant. Z value is calculated as 40.13 with the significant 0.00 for one-tailed test.

As a result, basket ratio of Week 1 can be assumed as equal with basket ratio of Week 3. On the other hand, basket ratio of Week 2 is significantly higher than both Week 1 and Week 3. That is, the recommendation system has increased basket ratio.

4. Discussion

This study has represented a recommendation engine by using association rules. The system had been tested by existing data in terms of the accuracy and the coverage. Best results have determined for 25 days data with 87.74% coverage and 16.43% accuracy. Processing time was 318 minutes for those data.

The deployment stage for an e-commerce website has taken three weeks. First week, there were 153322 searches and the number of products added to the cart was 23467. Second week, there were 158232 searches and the number of products added to the cart was 32878. Third week, there were 162934 searches and the number of products added to the cart was 24989. Basket ratios for each week was 15,31%, 20,78%, 15,34% respectively.

Standard normal distribution (z) test is progressed for testing statistical significance of differences for each week. It was seen that recommendations generated by association rules have significantly increased the basket ratio while there is no statistically significant difference between situations that recommendation system was inactive and that recommendations were generated randomly.

References

Agrawal R., Srikant R., Fast Algorithms for Mining Association Rules, *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994

Giudici P., *Applied Data Mining*, John Wiley & Sons, 2003

Han J., Kamber M., *Data Mining: Concepts and Techniques*, Morgan Kaufman, 2001

Hand D., Mannila H., Smyth P., *Principles of Data Mining*, The MIT Press, 2001

Hegland M., Algorithms for Association Rules, *Lecture Notes in Computer Science* (2600), 2003

Hipp J., Güntzer U., Nakhaeizadeh G., Algorithms for Association Rule Mining: A General Survey and Comparison, *SIGKDD Explorations*, Vol.2, No.1, 2000

Kotsiantis S., Kanellopoulos D., Associations Rule Mining: A Recent Overview, *GESTS International Transactions on Computer Science and Engineering*, Vol.32 No.1, 2006

Melville P., Mooney R., Nagarajan R., Content-Boosted Collaborative Filtering, *Proceedings of the SIGIR2001 Workshop on Recommender Systems*, New Orleans, 2001

Mobasher B., Dai H., Nakagawa M., Effective Personalization Based on Association Rule Discovery from Web Usage Data, *3rd ACM Workshop on Web Information and Data Management*, Atalanta, 2001

Mobasher B., Data Mining for Web Personalization, *Lecture Notes in Computer Science* (4321), 2007

Tan P., Steinbach M., Kumar V., *Introduction to Data Mining*, Addison Wesley, 2005