Web Usage Association Rule Mining System

Maja Dimitrijević
The Advanced School of Technology, Novi Sad, Serbia
dimitrijevic@vtsns.edu.rs

Zita Bošnjak
The University of Novi Sad, Faculty of Economics Subotica, Serbia
bzita@eccf.su.ac.yu

Abstract

Web usage log files generated on web servers contain huge amount of information that can be used for discovering web usage association rules, which can potentially give useful knowledge to the web usage data analysts. Association rule over-generation is a common problem in association rule mining that is further aggravated in web usage log mining due to the interconnectedness of web pages through the website link structure.

We implemented a system for the discovery of association rules in web log usage data as an object-oriented application and used it to experiment on a real life web usage log data set.

We proposed to alleviate the problem of web usage association rule over-generation by pruning the rules that contain directly linked pages out of the rule set. Our experiments showed that interestingness measures can successfully be used to sort the discovered association rules after the pruning method was applied. Most of the rules that ranked highly according to the interestingness measures proved to be truly valuable to a web master.

We compared confidence and lift interestingness measures and found that lift outperformed confidence, but only after the minimum confidence threshold was taken into account.

Keywords: association rule discovery system, web usage data, interestingness measures, website link structure

Introduction

Web usage log files generated on web servers contain huge amount of information suitable for applying data mining methods to discover potentially useful knowledge (Kosala & Blockeel, 2000; Wang, Li, & Zhang, 2005). Discovering web usage association rules is one of the popular data mining methods that can be applied on the web usage log data. The information contained in association rules can be used to learn about website visitor behaviour patterns, enhance website structure making it more effective for the visitors, or improve web marketing campaigns (Anand, Mulvenna, & Chavielier, 2004; Cooley, Mobasher, & Srivastava, 1997).

Originally, association rule mining algorithms were applied to the analysis of transactional databases (Agrawal, Imielinski, & Swami, 1993; Brin, Motwani, Ullman, & Tsur, 1997). When evaluating the association rule interestingness, various measures can be used to help find the rules that give maximally

Editor: Eli Cohen

useful information to the user (Geng & Hamilton, 2006; Tan, Kumar, & Srivastava, 2004). Some of the proposed association rule interestingness measures are a*ll-confidence* (Omiecinski, 2003), *collective strength* (Aggarwal & Yu, 1998), *conviction* and *lift* (Brin et al., 1997).

While association rule finding algorithms are complete in that they find all rules that satisfy defined constraints, they often result in a huge set of rules that is difficult to exploit in order to find those rules that are truly interesting to the user (Liu, Hsu, & Ma, 1999). This problem is aggravated in association rule mining of the web usage log data (Huang, Cercone, & Aijun, 2002).

Web usage data is specific and differs from the market basket data in the way that it contains a large number of tightly correlated items (web resources or web pages) due to the link structure of a website. Web pages that are tightly linked together often occur in the same visitor sessions, which is a reason that the generated set of association rules contains a huge number of so called "hard" association rules, which are not truly interesting to a user (Huang, 2007).

There are freely available data mining software systems that can be used for discovering association rules in the web log usage data (Weka 3, n.d.). Weka is a great data mining tool with a wide range of features (Witten, Frank, & Hall, 2011).

We used Weka in our previous research (Dimitrijevic & Bosnjak, 2010), but ran into several obstacles. We found that Weka did not contain an integrated set of tools to support all phases of web usage mining. A special software had to be used for web log preprocessing, such as WUM Prep scripts. Further, we had to develop our own special purpose converters of the web log preprocessed data into the Weka's ARFF file format before running web usage association rule discovery algorithm in Weka.

Other limitations of the current Weka version are that it offers four association rule interestingess measures without option of combining them. Weka can be extended to include more interestingness measures, but it would involve additional work. There is also no support for automatic pruning of the discovered association rules in Weka using the methods we proposed (this was confirmed to us by the Weka architect in a forum).

We concluded that it would give us more flexibility in our research and ability to apply the methods we propose for pruning association rules and for combining various interestingness measures if we used our own independent tool specialized for Web Usage Mining.

We present the first version of this tool in this paper. We apply the system for the discovery of the association rules on a real life data set and present the results using various parameter values.

We propose a method to alleviate the problem of over-generation of not truly interesting rules in web usage mining by eliminating the rules that contain directly linked pages. While this method does not completely solve the problem, our experiments show that it substantially alleviates it, leaving most not truly interesting rules out of the generated rule set.

The rest of the paper is organized as follows. We introduce definitions of the web usage association rule mining terms used in this paper in the section Important Definitions. The section Software Implementation gives an overview of our web usage association rule discovery system and explains the main ideas of the implementation. The section Pruning Directly Linked Pages explains our method for pruning irrelevant association rules. The section Results Analysis explains the steps of the association rule discovery process in our experiments and compares the performance of the rule interestingness measures we used. Finally, the section Conclusion gives an overview of the results.

Important Definitions

In this section we give a review of important definitions commonly used in web usage association mining, which we frequently refer to in the remainder of the paper.

Item

In the context of web usage association rule mining an item is a web resource of a particular website that can be requested by the website visitors.

Candidate

A candidate is a set of web resources contained in a particular website.

Session

We refer to a session as a set of web resources requested during a website visit.

It is hard to define session accurately. When a website visitor browses through a website, then makes a pause and returns, her/his visit may be considered as one or two sessions (Anand et al., 2004; Dettmar, 2004).

In this work we define a session as the set of web resources requested from the same IP address, where the time between two consecutive requests does not exceed 5 minutes.

Data set

We refer to the data set in terms of the website usage data mining as a set of sessions that contain the requests to web resources of a particular website.

Candidate support

Support of a candidate C is a measure of how frequently all items in the candidate C occur together in the set of all sessions of a data set. It is the probability of the set of items in C occurring together in a session in the data set. It is calculated as the count of sessions that contain the set of items C divided by the count of all sessions in the data set.

Frequent candidate

A candidate X is frequent if its support is higher than the user specified minimum support threshold.

For example, if a user sets minimum support threshold to 0.05, it means that a candidate is frequent if and only if at least 5% of all sessions contain all web resources that comprise the candidate.

Web usage association rule

Web usage association rule is a correlation between two candidates X and Y in the form

$X \rightarrow Y$, where $X \cap Y = \varnothing$

The candidate X is called the rule antecedent. The candidate Y is called the rule consequent.

Association rule support

The support of a rule $X \rightarrow Y$ is the support of the set $X \cup Y$.

Web usage association rule interestingness

The interestingness of an association rule refers to the practical usefulness of the knowledge discovered by the association rule data mining to a data analyst.

In this paper we refer to the interestingness of a discovered association rule as the likelihood that a web analyst would use the association rule to improve the structure of the website that is being analyzed.

Web usage association rule interestingness measure

An interestingness measure is a function that assigns a value to each association rule, which corresponds to the web usage rule interestingness.

Various interestingness measures can be used when the association rules are discovered to qualify the potential interestingness of the association rules and highlight potentially interesting rules for the data analyst (Geng & Hamilton, 2006; Omiecinski, 2003).

Confidence

Confidence is an interestingness measure of an association rule that refers to conditional probability of the rule consequent given the rule antecedent.

The rule $X \rightarrow Y$ holds in a set of sessions D with confidence c if c% of sessions in D that contain X also contain Y. If Supp denotes candidate support, the following formula can be used to calculate the confidence of a rule.

$Conf(X \rightarrow Y) = Supp(X \cap Y) / Supp(X)$

Lift

Lift is an interestingness measure of an association rule that compares the rule confidence to the expected rule confidence.

The expected confidence of a rule $X \rightarrow Y$ in a set of sessions D is the probability of the rule consequent Y in D. If the probability P(Y) is equal to the conditional probability P(Y|X), the item sets X and Y are not correlated in D.

If Supp denotes candidate support, the following formula can be used to calculate the lift of the rule $X \rightarrow Y$.

$Lift(X \rightarrow Y) = Conf(X \rightarrow Y) / Supp(Y)$

Web usage association rule mining

The problem of mining association rules is to generate a set of potentially interesting association rules in a data set of sessions that have support higher than the specified minimum support threshold and assign an interestingness value to all rules based on an interestingness measure.

Software Implementation

For the purpose of this research, we developed an independent Windows desktop application for the association rule discovery in web log data. The application was implemented using C# programming language. It is an object-oriented application specialized for the discovery of association rules in web log usage data, rather than in the general relational database data.

In a previous research (Dimitrijevic & Bosnjak, 2010) the open source (Weka 3) data mining software was used for discovering association rules in web log data. However, Weka does not support web log mining in an efficient and natural way, while it is better suited for relational database mining. Special purpose converters of the web log data into the Weka's "arff" file format had to be designed before it was possible to run association rule discovery algorithm in Weka.

Our web usage association rule discovery software is designed in a way that allows adding various interestingness measures to the rule discovery algorithm. Those measures can then be used

for evaluating interestingness of the generated association rules, in order to find the rules most valuable to a web analyst.

The following section describes main classes and their role in our web usage association rule discovery software.

Main Classes

The class *Item* contains the item url and the number of sessions in the input web log file that contain the item url request.

The list of urls contained in a session is contained in an object of the class *Session*. A list of all sessions is stored in memory in an object of the class *SessionTable*. This object is created once, at the beginning of the application run, when parsing the web log file.

An input web log file is read and parsed using various methods of the class *App*. This class is also a holder for the objects *ItemTable* and *SessionTable* that contain the list of all items and the list of all sessions respectively.

An object of the class *Candidate* contains the list of items that make up a candidate and the candidate support. To count the support for each candidate a pass through the data set is required. We plan to optimize this process in future and reduce the number of required passes through the data set.

An object of the class *Rule* contains the antecedent and the consequent candidates of the rule including their support, as well as the values of the various rule interestingness measures. The class also contains methods for calculating all the rule interestingness measures. It is easily extendible and additional interestingness measures can be added in future.

Input File

The data set is loaded from a textual web log file containing web log data cleaned of irrelevant and automatic requests and divided into sessions. The input file used for the experiment in this research can be found at http://www.vtsns.edu.rs/maja/insite2011.

Algorithm Output

Association rule discovery method based on the Apriori algorithm (Agarwal et al., 1993) consists of two phases. In the first phase, all frequent candidates are generated level-wise. At the end of this phase, the candidates are listed in a textbox in the user interface form.

In the second phase of the association rule discovery algorithm, the rules are generated based on the frequent candidates. The rules are written in the form of a table to a CSV file. Each row in this file represents one association rule. The first and second columns contain the list of urls of the rule antecedent and the rule consequent respectively. The following three columns contain the support of the antecedent and the consequent of the rule and the support of the rule as a whole, respectively. There can be several columns at the end, each of which contains the values of various interestingness measures of a rule. For the purpose of this research we used two interestingness measures: Confidence and Lift.

Candidate Generation

We implemented a version of Apriori algorithm to generate candidates level-wise. Each candidate is stored in an object of the class *Candidate*. A candidate is considered frequent if its support exceeds the threshold value selected by the user. We call the list of all frequent candidates of length N the *level N*, and store it in an object of the *CandidateLevel* class.

At the end of the generation of a candidate level, we sort candidates within each level alphabetically according to their item urls. We use string representation of each candidate created by appending all urls of the candidate in their alphabetical order. Based on the sorted candidate list we can efficiently find each candidate and its support during the association rule generation phase using binary search.

Rule Generation

In the association rule generation phase we generate the rules level-wise based on the rule length, where rule length is the number of pages in the rule. When creating rules of length N, we use the list of candidates at the level N. For each candidate, we generate all possible antecedents and consequents of an association rule by applying a subset generation algorithm on the candidate of length N.

We use the values of the antecedent and consequent support to calculate the rule interestingness measures. Since all candidate levels are sorted in the frequent set generation phase, we apply binary search to find the rule antecedent and consequent and their support in one of the previously generated candidate levels.

There are no passes through the data set when generating the rules, which makes the process very efficient.

At the end of rule generation all rules, including the urls of the left and right hand side, the support and all rule interestingness measures are written to a CSV file.

User Interface

Figure 1 shows the user interface of the software. When an input file containing web log usage data is opened and parsed, the list of all items and the number of sessions that contain the items are written to the Items list box on the left hand side of the form.

A user selects the support and confidence threshold, as well as the maximum levels to generate, which are the parameters used in the frequent set generation.

A user starts the Apriori frequent set generation algorithm by clicking the “Generate Frequent Sets” button. The generated frequent sets are shown in the list box on the right hand side of the form.

The last phase of the rule discovery is initiated by clicking the “Generate Rules CSV Format” button, which starts the rule generation algorithm. The rules are then written to a CSV file.

Pruning Directly Linked Pages

When association rule mining is applied to the web log usage data, a huge number of rules with extremely high confidence is generated. Those “hard” rules do not truly reflect website visitor interests, but are natural consequences of the hyperlinks that exist between the pages comprising the rules. Such rules are not valuable to a web analyst and should be pruned out of the rule set (Kannan & Bhaskaran, 2009).

In Dimitrijevic and Bosnjak (2010) pruning techniques based on the schemes for rule clustering and removing irrelevant rules were used. We now propose a rule pruning method based on avoiding rules that contain directly linked pages.

Definition 1: Directly linked pages

We say that two pages a and b are *directly linked* if there is a hyperlink from page a to page b in the body text on the page a, or there is a link from page b to page a in the body text on the page b.

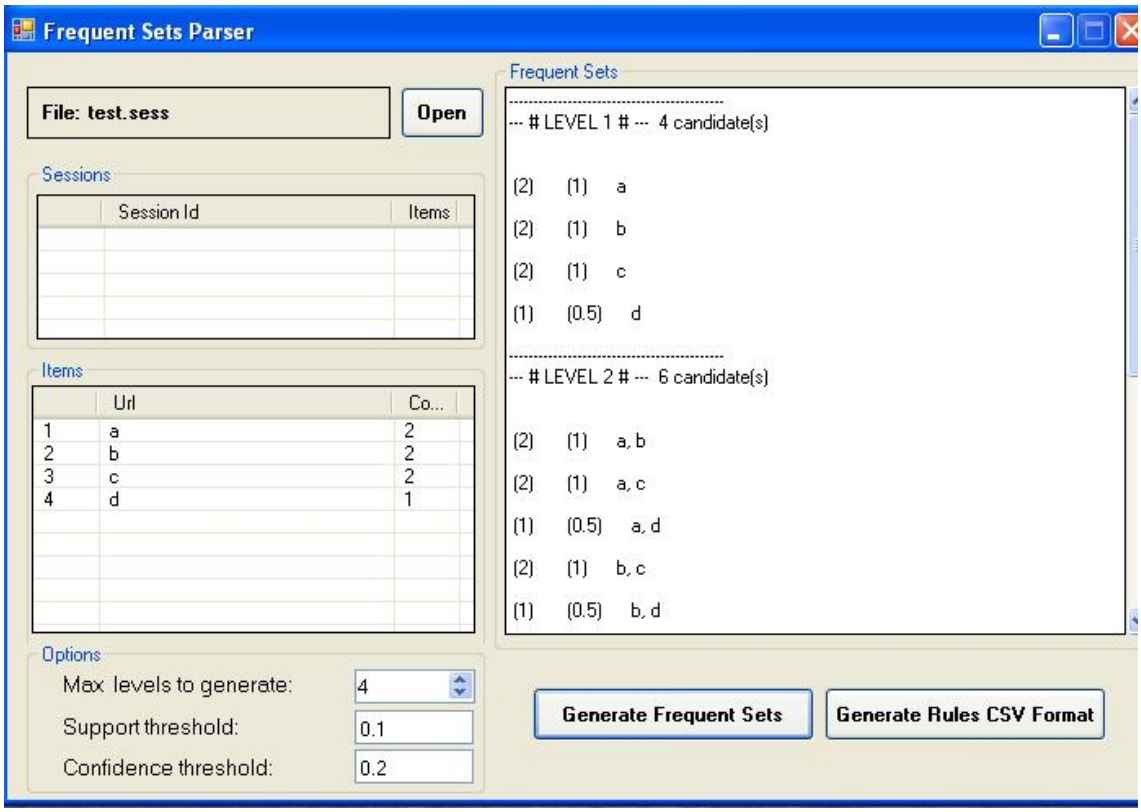

Figure 1: A glimpse of the user interface

This excludes the links through a main website menu, which exists either on the top, bottom or on the side of the web pages, and is shared among most web pages of the site.

Definition 2: Rules containing directly linked pages

Let X be the consequent and Y the antecedent of an association rule "X --> Y". We say that the rule contains a pair of directly linked pages if and only if the set X ∪ Y contains two directly linked pages.

Pruning association rule set

We propose to prune all rules that contain a pair of directly linked pages out of the association rule set generated by the association rule discovery algorithm. We have conducted experiments on a real data set in order to test the validity of this method.

This method is an attempt to alleviate the association rule over-generation in mining web log usage data caused by hard connectedness of items (web pages). The method does not fully eliminate this problem, but it substantially alleviates it.

This pruning process can be embedded within the association rule discovery algorithm (frequent set discovery phase), which would substantially increase its efficiency. We leave the implementation, as well as comprehensive testing of this method for future work.

Association rule true interestingness

To evaluate the usability of the interestingness measures, we compare their values to the true interestingness of association rules, according to Definition 1. The definition of the rule true inter-

estingness is based on how likely it is that a web master would act upon the knowledge conveyed by the association rule.

Definition 3: Web usage association rule true interestingness

We consider a web usage association rule truly interesting if it is expected to cause a web master to take an action to change the structure of the website, based on the knowledge acquired through the association rule.

We define three categories of the rule true interestingness according to Definition 3. The rules in the first category are those that are expected to cause a web master to take an action to change the website structure. The rules in the second category are those based on which a web master might take such action. The rules in the third category are those that are not expected to cause a web master to act.

Experiment Results

Data Set

We have conducted the experiments on the web log file containing information about all web requests to the official website of the Advanced School of Technology in Novi Sad made in the month November 2010. The file contains 397,741 web requests and can be found at http://www.vtsns.edu.rs/maja/insite2011. The data set is about 30 times larger than that used in Dimitrijevic and Bosnjak (2010). Our association rule discovery software proved to be efficient and completed each test run within seconds.

Data Preparation

The first step in the web log mining process is to clean web log data of irrelevant and automatic web requests prior to running association rule discovery algorithm, as well as to group web requests into visitor sessions (Anand et al., 2004).

For our experiments, we used a freely available tool for web log data preparation called WumPrep which consists of a set of Perl scripts for cleaning the web log file of irrelevant and automatic requests and creating sessions in it (Dettmar, 2004).

The cleaning process reduced the number of web requests down to 93,688 relevant requests. The requests are grouped into 16,637 sessions of the website visitor made in the month of November 2010.

Data preparation using WUMPrep scripts is a straightforward and efficient one time procedure that prepares the data, which can then be used for the association rule discovery many times while selecting the optimum parameter values for the algorithm. Therefore, we found no need to implement our own data preparation into our association rule discovery software.

Result Analysis

After the data preparation phase, the web usage log file is loaded into our association rule discovery system. The association rules are then generated in two phases: Frequent set generation and Association rule generation.

Frequent set generation and support threshold

Prior to running the frequent set generation phase of the association rule discovery algorithm based on the Apriori algorithm, the minimum support threshold parameter needs to be set to the value that gives optimal results. If the support threshold is set too low, too many potentially not

truly interesting rules are generated, cluttering the rule set and making it hard to understand for the final user. On the other hand, if the support threshold is set too high, there is a chance that too many potentially interesting rules are missed from the rule set.

In our experiments, we found that support values higher than 0.01 generate too few rules, leaving many potentially interesting rules out of the rule set. We conducted three experiments setting the minimum support threshold to 0.007, 0.0085 and 0.01 and compared the resulting association rules and their interestingness.

In each test, the execution time of the frequent set generation phase was within the range of a few seconds up to two minutes. There is room to optimize the software and lower the execution time, but we leave that for future work.

Association rule generation and confidence threshold

In the second phase, our software generates potentially interesting association rules and calculates their interestingness measures, based on the frequent item sets and their support.

One of the basic association rule interestingness measures is confidence, and we consider its minimum threshold when evaluating the value of the calculated rule interestingness. We found that in all our experiments the rules whose confidence was lower than 0.4 were extremely rarely truly interesting according to our definition. Therefore, we chose to set the minimum confidence threshold to 0.4 and prune out all the rules that have lower confidence. We found that no rules that are expected to cause a web master to act according to our definition were pruned out from the rule set in this process.

The association rule generation phase took milliseconds in all of our experiments. The resulting CSV file was generated momentarily as the user would initiate rule generation.

Evaluating rule interestingness

We use Confidence and Lift as two different interestingness measures of the generated association rules and compare their values to the true rule interestingness based on the Definition 3.

We asked the web master of the Advanced School of Technology in Novi Sad website to categorize the association rules discovered in our experiments into the three given rule interestingness categories (“expected to cause an action”, “might cause an action”, “not expected to cause an action”) for the purpose of our experiments. We counted the number of rules in each true interestingness category when the rules are sorted according to Confidence and Lift interestingness measures.

Similar approach was taken in (Huang et al., 2002) where top 10 rules were classified as interesting or not interesting by a domain expert. However, almost half of their rules had confidence close to 1, while many of them were not truly interesting to the domain expert. This was most likely due to the connectedness of the pages within the rules through the website link structure.

Our results are more stable, while the rule set is not cluttered by too many not truly interesting rules. By eliminating the rules containing directly linked pages according to Definition 2, we eliminated almost all rules that had confidence close to 1 in our experiments.

We show that both Confidence and Lift perform satisfactory, while Lift somewhat outperforms Confidence in all our experiments.

We present the results of the three association rule discovery runs, using different support values in the following sections.

Interestingness based on Confidence

We sorted the discovered association rules according to Confidence only and show how the top 10 rules are distributed over the three categories of true rule interestingness.

Each column in Table 1 corresponds to an association rule algorithm run where the minimum support threshold is set to the specified value. Each cell in the table contains the number of rules that fit into the rule interestingness category according to Definition 3.

Table 1: True interestingness of the top 10 rules according to Confidence

	Supp=0.007	Supp=0.0085	Supp=0.01
Expected	6	6	6
Might	4	3	3
Not expected	0	1	1

In Figure 2 we present the data in the histogram format, with the rule interestingness categories listed on the side. The vertical axis corresponds to the number of top 10 rules in each category.

Figure 2: Top 10 rules according to Confidence

When the rules are sorted according to confidence, most of the top 10 rules are expected to, or might cause the web master to act in all three experiments with different support values.

Interestingness based on Lift

We sorted the discovered association rules according to Lift only and show how the top 10 rules are distributed over the three categories of true rule interestingness.

Each column in Table 2 corresponds to an association rule algorithm run where the minimum support threshold is set to the specified value. Each cell in the table contains the number of rules that fit into the rule interestingness category according to Definition 3.

Table 2: True interestingness of the top 10 rules according to Lift

	Supp=0.007	Supp=0.0085	Supp=0.01
Expected	9	9	7
Might	1	1	3
Not expected	0	0	0

Figure 3 presents the distribution of the top 10 rules according to Lift, over the true interestingness categories. The vertical axis corresponds to the number of top 10 rules in each category.

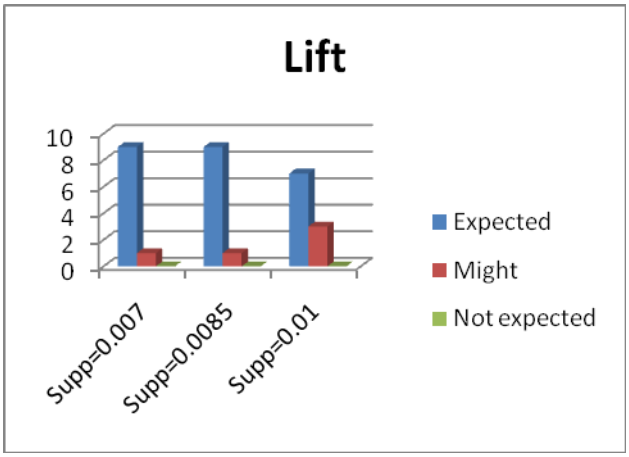

Figure 3: Top 10 rules according to Lift

When the rules are sorted according to lift almost all of the top 10 rules fit into the first category of the true interestingness in all three experiments with different support values. There were no rules in the top 10, based on which a web master would not be expected to act in all three experiments.

Optimal interestingness measures

In our experiments, minimum confidence threshold is set to 0.4 and the rules with lower confidence are pruned out of the rule set prior to sorting all remaining rules according to either lift or confidence. We showed that Lift then gives better results than Confidence with respect to the true rule interestingness according to the Definition 3. Lift almost perfectly accurately measures interestingness in our experiments and no other measure needs to be considered.

However, in our initial experiments when the confidence threshold had been set lower (for example 0.2), lift alone did not perform that well. Neither Confidence nor Lift proved to be reliable in determining true interestingness of the association rules under these conditions.

We conclude that Lift and Confidence need to be considered in combination when deciding on the association rule interestingness. In this research we decided to combine them by assigning minimum confidence threshold value of 0.4 before applying Lift as the rule interestingness measure. While this way of combining Lift and Confidence gave excellent results in our experiments, it would be worth conducting experiments on other web log usage data sets in order to re-evaluate this method, or find other ways to use web usage association rule interestingness measures. We leave this direction of research for future work.

Conclusion

We have implemented a system for the discovery of association rules in the web log usage data as an object-oriented software application easily extendible by various rule interestingness and pruning modules. The initial tests were conducted on real life web usage log data containing 93,688 relevant requests grouped into 16,637 sessions. The system performed well in the initial tests.

Too many rules that are not truly interesting to a web analyst often have extremely high confidence values when the association rule generation algorithm is applied. Such rules are hard to interpret and cannot give useful information about website visitor interests.

We alleviated the problem of web usage association rule over-generation by pruning the rules that contain directly linked pages out of the rule set. Our experiments showed that interestingness measures can successfully be used to sort the discovered association rules after applying the pruning method. Most of the rules that ranked highly according to the interestingness measures proved to be truly valuable to a web master.

We compared the performance of confidence and lift interestingness measures and found that lift outperformed confidence, but only after a minimum confidence threshold was taken into account.

Directions for Future Work

While our method gave excellent results on our experimental data set, more tests should be performed on different website usage data sets to confirm the results.

Our system implementation can be optimized by reducing the number of required passes through the data set. This may be particularly useful for sparse data, where candidates do not occur in too many sessions. That is the case when looking for the rules that have low support but are potentially interesting to a web master.

The purpose of this research was to perform initial tests of the system validity and performance. We plan to perform more extensive system efficiency testing in our future research.

We plan to extend our system in the future by adding rule pruning modules, as well as implementing other interestingness measures in order to fine tune the results.

The rules that contain directly linked pages may in some cases be valuable to a web master, such as to learn about usefulness of the outgoing hyperlinks on a web page. However, other methods than association rule discovery should be devised for that purpose. We leave such analysis out of the scope of this paper.

References

Aggarwal, C. C., & Yu, P. S. (1998). A new framework for itemset generation. In *PODS 98, Symposium on Principles of Database Systems*, 18-24.

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 207-216.

Anand, S. S., Mulvenna, M., & Chavielier, K. (2004). On the deployment of web usage mining. In B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, & G. Stumme (Eds.), *Web mining: From web to semantic web* (pp. 23-42). Berlin: Springer.

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 255-276.

Cooley, R., Mobasher, B., & J. Srivastava. (1997). Web mining: Information and pattern discovery on the World Wide Web. *Proc. IEEE Intl. Conf. Tools with AI*, 558-567.

Dettmar, G. (2004). *Logfile preprocessing using WUMprep*. Talk given at the Web Mining Seminar in Winter semester 2003/04, School of Business and Economics, Humboldt University Berlin, Berlin.

Dimitrijevic, M., & Bosnjak, Z. (2010). Discovering interesting association rules in the web log usage data. *Interdisciplinary Journal of Information, Knowledge, and Management, 5,* 191-207. Retrieved from http://www.ijikm.org/Volume5/IJIKMv5p191-207Dimitrijevic443.pdf

Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR), 38*(3).

Huang, X. (2007). Comparison of interestingness measures for web usage mining: An empirical study. *International Journal of Information Technology & Decision Making (IJITDM), 6*(1), 15-41.

Huang, X., Cercone, N., & Aijun, A. (2002). Comparison of interestingness functions for learning web usage patterns. *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA*, November 4-9, 2002, 617-620.

Kannan, S., & Bhaskaran, R. (2009) Association rule pruning based on interestingness measures with clustering. *International Journal of Computer Science Issues, IJCSI, 6*(1), 35-43.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *SIGKDD Explorations, 2*(1), 1-15.

Liu, B., Hsu, W., & Ma, Y. (1999). Pruning and summarizing the discovered associations. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 125-134.

Omiecinski, E.R. (2003). Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering, 15*(1), 57-69.

Tan, P., Kumar, V., & Srivastava, J. (2004). Selecting the right interestingness measure for association patterns. *Information Systems, 29*(4), 293-313.

Wang Y., Li, Z., & Zhang, Y. (2005). Mining sequential association-rule for improving Web document prediction. *Computational Intelligence and Multimedia Applications*, 146-151.

Weka 3: Data Mining Software in Java. (n.d.). University of Waikato. Retrieved October, 2010, from http://www.cs.waikato.ac.nz/ml/weka/

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data mining: Practical machine learning tools and techniques (3rd ed.). Morgan Kaufmann. Available from http://www.cs.waikato.ac.nz/ml/weka/book.html

Biographies

Maja Dimitrijević is a lecturer at the Advanced School of Technology in Novi Sad. She teaches database structures, object-oriented programming and software engineering. She is currently working on her PhD thesis in the area of data mining. Her current research interests include data mining, web usage mining, database structures and software engineering. She holds an MSc degree in Computer Science from the University of British Columbia, Vancouver, Canada.

Zita Bošnjak is a full professor at the University of Novi Sad, Faculty of Economics Subotica, Department of Business Information Systems and Quantitative Methods. She received a B.S. (1987) in Informatics from the University of Novi Sad, Faculty of Sciences and an M.S. (1991) and a Ph.D. (1995) in Informatics from the University of Novi Sad, Faculty of Economics Subotica. Her current research interests include the theory and practice of knowledge in data discovery and expert and fuzzy systems, and their application to business, strategic management, education and capacity building. She has written over 20 journal articles, 3 books, and 50 conference articles on related topics. From January 2006 she has been a member of the editorial board of the *Management Information Systems* journal.