Hierarchical frequent pattern analysis of web logs for efficient interestingness prediction

Sudhamathy G1,* , Jothi Venkateswaran C2

1Department of Computer Applications, Velammal College of Engineering & Technology, Madurai 625 009, India, Email: sudhamathi10@hotmail.com
2Department of Computer Science, Presidency College (Autonomous), Chennai 600 025, India, Email: jothivenkateswaran@yahoo.co.in

*Corresponding author: Department of Computer Applications, Velammal College of Engineering & Technology, Madurai 625 009, India, Email: sudhamathi10@hotmail.com

Abstract

In this paper, we proposed an efficient approach for frequent pattern mining using web logs - web usage mining and we call this approach as HFPA. In our approach HFPA, the proposed technique is applied to mine association rules from web logs using normal Apriori algorithm, but with few adaptations for improving the interestingness of the rules produced and for applicability for web usage mining. We applied this technique and compared its performance with that of classical Apriori-mined rules. The results indicate that the proposed approach HFPA not only generates far fewer rules than Apriori-based algorithms (FPA), but also generate rules of comparable quality with respect to three objective performance measures namely, Confidence, Lift and Conviction. Association mining often produces large collections of association rules that are difficult to understand and put into action. In this paper we have proposed effective pruning techniques that were characterized by the natural web link structures. 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 site administrator.

Key words: Web usage mining, Web logs, Association rules, Interestingness measures

Abbreviation: HFPA - Hierarchical frequent pattern analysis.

Introduction

Originally, association rule mining algorithms were applied for market basket analysis which contained transaction data. The transaction data may include many records of which each record has a transaction id and a list of items purchased during that transaction. But when the same Apriori algorithm has to be applied for web log data, it has to be transformed to the same format as that of the transactions (Kannan & Bhaskaran, 2009). To make this happen, the web log data has to be cleaned, split and preprocessed into sessions and the list of web pages navigated during each session. Once this data transformation is done, association rules can be mined as we do for market basket analysis (Eirinaki & Vazirgiannis, 2000). However, the threshold selection, pruning method, interesting measures used and ranking of the rules needs some modifications to suit the needs of web usage mining.

The association rule mining algorithm can find all rules that satisfy defined constraints, they often result in a large set of rules that is difficult to exploit and find those rules that are truly interesting to the user (Iváncsy & Vajk 2008). Web log data differs from the market basket data in the sense that it contains a large number of tightly correlated web pages due to the link structure of a website (Han& Elmasri, 2004). Web pages that are tightly linked together often occur in the same transaction, which is why the generated set of association rules are high and they have very high confidence, but are not truly interesting to the user (Huang, 2007). So, in our approach we prune the item set that includes directly linked pages as we are only interested in the information that can prompt actions leading to enhancement of a website and improving the browsing experience for visitors. As a case study to prove the efficiency of the proposed approach we used the web log files of www.eretailstore.biz for a
period of six months, 07/2010 to 12/2010. These web log files were cleaned, preprocessed, transformed to match the format suitable for Apriori algorithm. We set the minimum threshold for support and confidence that will suit the web logs analysis. After acquiring the rules from the frequent item sets produced by Apriori algorithm, we calculated support, confidence, lift and conviction. We sorted by descending order of lift value and then by descending order of conviction. The top ranked rules, say top 20% of the overall rules produced after pruning and threshold limit exceeding are found to be the most useful and interesting rules that can recommend better web site reorganization or personalization.

The approach is compared with the traditional approach of using just support and confidence to find the interesting rules as in the case of market basket analysis. It was found that the proposed approach HFPA is performing better with respect to computation time, number of rules produced finally, percentage of accuracy of interestingness, memory usage and the number of rules reduction by applying the pruning techniques.

Proposed approach

Overview of the HFPA is represented pictorially by the Fig. 1. This approach includes four main steps, namely, preprocessing, frequent item sets generation, rules generation and measuring interestingness.

Pre-Processing

The web log files are extracted from the web server and it is preprocessed. The preprocessing includes the steps of parsing, cleaning and session identification. The preprocessing step is executed for each web log file at a time.

Actually the web log files are flat text files that contain many space / tab delimited fields. The important fields in any web log file are data, time, client IP address, server IP address, server port, URL visited and user agent filed that gives details of the browser and operating system versions. The parsing step does splitting of the text file into specific fields and extracts the required fields into a database table. In our case, we need the fields, date, time, client IP address, URL visited and user agent. Once these fields are split, extracted and stored in a database table, the extracted records are then cleaned to remove the images, icons and unwanted requests. So we deleted all records that have .JPEG, .GIF and .CSS files in the URL visited field. As a result we have the cleaned database with relevant records. The next step in preprocessing is user session identification. If we observe the log entries in the web log, files are chronologically ordered based on the different user’s requests from their client machine to the web server.

The user sessions are identified based on the following assumptions: firstly, each user has a unique client IP address while browsing the website. The same IP address can be assigned to other users after the user finishes browsing. Secondly, the couple of client IP address and user-agent are considered for single user identification as different users can come from the same proxy. Finally, any user accessing the website from a unique client IP address and user-agent details will be active in a session only if he does not exceed the maximum idle time. This maximum idle time is optimally 5 minutes. [That is if the difference between the time of the web log entries from the same client IP address and user agent field is more than 5 minutes then both the log entries belong to different user sessions.]
Thus one user session is the set of records that have the same client IP address and user agent for which the consecutive date and time does not exceed the idle time of 5 minutes.

**Frequent Item Sets Generation**

Frequent Item sets can be generated by applying the enhanced version of Apriori algorithm, but before applying this algorithm we need to set up the minimum threshold for support and confidence. The ideal minimum threshold for support is set to be 10% (or 0.1) and for confidence is set to be 45% (or 0.45) for web usage mining.

Before going into the Apriori algorithm, we need to understand two properties of the algorithm. Firstly, the Apriori property of “Any subset of frequent item set must be frequent”. Example - if \{P1, P2\} is a frequent item set, then its sub sets \{P1\} and \{P2\} are also said to be frequent. Secondly, the join operation which states that “To find a set of candidate k item sets, it is generated by joining the frequent k-1 item sets with itself”.

Example - if we have the frequent 2 item sets \{P1, P2\} and \{P1, P3\}, then we get the candidate 3 item sets by joining \{P1, P2\} and \{P1, P3\}. The candidate 3 item set formed will be \{P1, P2, and P3\}. In the enhanced Apriori algorithm of HFPA, we consider \(C_k\) as the candidate item set of size k and \(L_k\) as the frequent item set of size k.

**Pseudo code for Enhanced Apriori Algorithm**

Let \(L_1 = \{\text{frequent 1 item sets}\}\);

For (k=1; \(L_k \neq \emptyset\); k++) do begin

Generate \(C_{k+1}\) from \(L_k\) using the join operation;

Initialize the count of all candidates in \(C_{k+1}\) to 0;

For each record r in the session database do

If (number of items in the record r \(\geq k+1\)) then

Increment the count of all candidates in \(C_{k+1}\) that are contained in r by 1;

End If;

End For;

\(L_{k+1}\) = Candidates in \(C_{k+1}\) with support greater than the minimum support threshold – Candidates in \(C_{k+1}\) that reflect the direct link structure of the web site.

End For;

Return \(L_1 \cup L_2 \cup \ldots L_k\)

There are two enhancements done to the normal Apriori algorithm in the above algorithm. Firstly, while scanning the session database for support count for a candidate item set, skip the records that have less number of items (or pages) than in the candidate item set. This is because the pages in the candidate item set will not be present in such records that have less number of pages than in the candidate item set. Secondly, prune the item sets that reflect the direct link structure of the web site from the frequent item set list. This is because the rules produced by these item sets will represent the website link correlation and hence are not truly interesting to the user. Now as a result of applying the above said algorithm we would have got the optimal frequent item sets from which next we generate the rules.

**Rules Generation**

Consider all the frequent item sets except the one frequent item sets for rules generation. Rules generation is done in two steps. As a first step, for each frequent item set I, generate all non empty subsets of I. As a second step, for every nonempty subset S of I, output the rule S \(\rightarrow\) I – S. Example - consider the frequent item set \{P1, P5\}. The sub sets are – \{P1\} and \{P5\}. Hence the rules are – \{P1\} \(\rightarrow\) \{P5\} & \{P5\} \(\rightarrow\) \{P1\}. In the above, note that the item on the left hand side of the rule is called the rule antecedent and the item on the right hand side of the rule is called the rule consequent. Next calculate the support and confidence for each of the rules. The support for all the rules will definitely be greater than the minimum support threshold as we have already pruned the item sets whose support is less than the minimum support threshold.
But many rules may exist whose confidence is less than the minimum confidence threshold. Hence we have to prune the rules whose confidence is less than the minimum specified confidence threshold. Also the rule whose confidence is equal to one is also pruned as they mean the strong correlation of pages due to link structure of the web site. These rules with confidence = 1 will not be interesting for the user. After pruning the unwanted rules based on confidence we have to calculate lift and conviction for the remaining rules. This is done as per the formulae mentioned in the definition above. Now we have a list of rules and their corresponding support, confidence, lift and conviction.

Measuring Interestingness

From the list of rules with their support, confidence, lift and conviction, we have to focus at the most interesting rules that can help a web site administrator to improve the web site. For this we need to sort the rules by descending order of lift and then by descending order of conviction and rank them accordingly. The top 20% of the overall pruned rules are found to be the most interesting and expected to be taken up for further action by the web master.

Performance Evaluation

The proposed approach of HFPA has to be evaluated and compared with the traditional market basket analysis way of association rule mining which we call it as FPA. To make this possible we developed a software using java that incorporated
all the steps and algorithm as proposed in the approach. As said before, we used the web log files of www.eretailstore.biz web site taken for a period of six months from 07/2010 to 12/2010 as a test data to evaluate the proposed approach against the traditional approach. Fig 2,3,4,5,6

Thus HFPA approach takes lesser run time, produces less number of rules from which interestingness have to be evaluated, gives high percentage of interestingness accuracy, uses less memory for processing and prunes more rules than the traditional frequent pattern analysis (FPA) approach. Thus this is a hierarchical frequent pattern mining approach that is found suitable for analyzing web log data and to predict useful information from the analyzed data.

The java software tool developed to do this performance evaluation also provides the opportunity for the users to set up different support and confidence threshold apart from what is suggested in this paper. By this way we can test the approach for different web sites log files. Also the tool allows the user to select the top ranking percentage by which we set the bottom line for selecting interesting rules. By this way we give the flexibility to get more interesting rules which might be found useful for the web master. The rules can be categorized as “Expected to cause an action”, “Might cause an action” and “Not expected to cause an action”. The domain expert can then take the necessary action accordingly.

Conclusion

One of the major problems in the domain of web usage mining is that, the size of association rules produced increases dramatically due to the existence of rules that have very high confidence because of the interconnectedness of web pages through the link structure. In order to deal with this issue of rule over-generation, we proposed pruning of item sets at the initial stage itself that causes such uninteresting rules. The HFPA approach is also beneficial since we can reduce the number of records scanned in the session database while counting for the frequent item sets. This may be particularly useful for sparse data, where candidates do not occur in too many sessions. Even though we have many interesting measures available for association rule mining, we have found the optimal minimum threshold for support and confidence and seen that the ordering of lift and conviction values suit ranking of rules formed out of web log files. The count of interesting rules considered by the web site administrator to improve the browsing experience of the users is left to the individual performing the analysis, even though we propose an optimal percentage of top ranked rules. We compared the HFPA approach with the traditional FPA approach and found that the new HFPA approach outperformed the existing FPA approach in many aspects like run time, memory usage, rules pruned, rules produced and accuracy percentage. There are scopes for future work in this proposal, like we can apply the same for different web sites to confirm the results and we can explore on other interestingness measures to see if they give better results.

References