International Journal of Soft Computing 3 (2): 99-105, 2008
ISSN: 1816-9503
© Medwell Journals, 2008

Discovering Frequent Patterns and Trends by Applying Web Mining Technology in Web Log Data

[1]C. Umapathi and [2]J. Raja
[1]Sathayabama University, Chennai 119, India
[2]S.S.N. College of Engineering,Chennai 119, India

Abstract: The expansion of the World Wide Web has resulted in a large amount of data that is now freely available for user access. The data have to be managed and organized in such a way that the user can access them efficiently. For this reason the application of data mining techniques on the Web is now the focus of an increasing number of researchers. One key issue is the investigation of user navigational behavior from different aspects. For this reason different types of data mining techniques can be applied on the log file collected on the servers. In this study 3 of the most important approaches are introduced for web log mining. All the 3 methods are based on the frequent pattern mining approach. The 3 types of patterns that can be used for obtain useful information about the navigational behavior of the users are page set, page sequence and page graph mining.

Key words: Pattern mining, sequence mining, graph mining, web log mining, automata

INTRODUCTION

The expansion of the World Wide Web has resulted in a large amount of data that is now in general freely available for user access. The different types of data have to be managed and organized in such a way that they can be accessed by different users efficiently. Therefore, the application of data mining techniques on the Web is now the focus of an increasing number of researchers. Several data mining methods are used to discover the hidden information in the Web. However, Web mining does not only mean applying data mining techniques to the data stored in the Web. The algorithms have to be modified such that they better suit the demands of the Web. New approaches should be used which better fit the properties of Web data. Furthermore, not only data mining algorithms, but also artificial intelligence, information retrieval and natural language processing techniques can be used efficiently. Thus, Web mining has been developed into an autonomous research area.

The focus of this study is to provide an overview how to use frequent pattern mining techniques for discovering different types of patterns in a Web log database. The 3 patterns to be searched are frequent itemsets, sequences and tree patterns. For each of the problem an algorithm was developed in order to discover the patterns efficiently. The frequent itemsets (frequent page sets) are discovered using the ItemsetCode algorithm presented in Ivancsy and Vajk (2005a). The main advantage of the Itemset Code algorithm is that it discovers the small frequent itemsets in a very quick way, thus the task of discovering the longer ones is enhanced as well. The algorithm that discovers the frequent page sequences is called SM-Tree algorithm (Ivancsy and Vajk, 2005) and the algorithm that discovers the tree-like patters is called PDTree algorithm (Batista *et al*., 2002). Both of the algorithms exploit the benefit of using automata theory approach for discovering the frequent patterns. The SM-Tree algorithm uses state machines for discovering the sequences and the PD-Tree algorithm uses pushdown automatons for determining the support of the tree patterns in a tree database.

WEB MINING TAXONOMY

Web mining involves a wide range of applications that aims at discovering and extracting hidden information in data stored on the Web. Another important purpose of Web mining is to provide a mechanism to make the data access more efficiently and adequately. The third interesting approach is to discover the information which can be derived from the activities of users, which are stored in log files for example for predictive Web caching (Yang and Zhang, 2003). Thus, Web mining can be categorized into 3 different classes based on which part of

Corresponding Author: C. Umapathi, Research Scholar, Sathayabama University, Chennai 119, India

the Web is to be mined (Kosala and Blockeel, 2000; Cooley *et al*., 1999). These 3 categories are, Web content mining, Web structure mining and Web usage mining. Web content mining (Li Shen *et al*., 2000) is the task of discovering useful information available on-line. There are different kinds of Web content which can provide useful information to users, for example multimedia data, structured (i.e. XML documents), semi-structured (i.e. HTML documents) and unstructured data (i.e. plain text). The aim of Web content mining is to provide an efficient mechanism to help the users to find the information they seek. Web content mining includes the task of organizing and clustering the documents and providing search engines for accessing the different documents by keywords, categories, contents etc. Web structure mining (Srivastava *et al*., 2000; Nanopoulos and Manolopoulos, 2001) is the process of discovering the structure of hyperlinks within the Web. Practically, while Web content mining focuses on the inner-document information, Web structure mining discovers the link structures at the inter-document level.

The aim is to identify the authoritative and the hub pages for a given subject. Authoritative pages contain useful information and are supported by several links pointing to it, which means that these pages are highly referenced. A page having a lot of referencing hyperlinks means that the content of the page is useful, preferable and maybe reliable. Hubs are Web pages containing many links to authoritative pages, thus they help in clustering the authorities. Web structure mining can be achieved only in a single portal or also on the whole Web. Mining the structure of the Web supports the task of Web content mining. Using the information about the structure of the Web, the document retrieval can be made more efficiently and the reliability and relevance of the found documents can be greater. The graph structure of the web can be exploited by Web structure mining in order to improve the performance of the information retrieval and to improve classification of the documents. Web usage mining is the task of discovering the activities of the users while they are browsing and navigating through the Web. The aim of understanding the navigation preferences of the visitors is to enhance the quality of electronic commerce services (e-commerce), to personalize the Web portals or to improve the Web structure and Web server performance. In this case, the mined data are the log files which can be seen as the secondary data on the web where the documents accessible through the Web are understood as primary data. There are 3 types of log files that can be used for Web usage mining. Log files are stored on the server side, on the client side and on the proxy servers. By having more than one place for storing the information of navigation patterns of the users makes the mining process more difficult. Really reliable results could be obtained only if one has data from all three types of log file. The reason for this is that the server side does not contain records of those Web page accesses that are cached on the proxy servers or on the client side. Besides the log file on the server, that on the proxy server provides additional information. However, the page requests stored in the client side are missing. Yet, it is problematic to collect all the information from the client side. Thus, most of the algorithms work based only the server side data. Some commonly used data mining algorithms for Web usage mining are association rule mining, sequence mining and clustering.

THE PROCESS OF WEB LOG MINING

Web usage mining, from the data mining aspect, is the task of applying data mining techniques to discover usage patterns from Web data in order to understand and better serve the needs of users navigating on the Web . As every data mining task, the process of Web usage mining also consists of 3 main steps:

- Preprocessing.
- Pattern discovery.
- Pattern analysis.

In this research pattern discovery means applying the introduced frequent pattern discovery methods to the log data. For this reason the data have to be converted in the preprocessing phase such that the output of the conversion can be used as the input of the algorithms. Pattern analysis means understanding the results obtained by the algorithms and drawing conclusions.

Figure 1 shows the process of Web usage mining realized as a case study in this work. As can be seen, the input of the process is the log data. The data has to be preprocessed in order to have the appropriate input for the mining algorithms. The different methods need different input formats, thus the preprocessing phase can provide three types of output data. The frequent patterns discovery phase needs only the Web pages visited by a given user. In this case the sequences of the pages are irrelevant. Also the duplicates of the same pages are omitted and the pages are ordered in a predefined order. In the case of sequence mining, however, the original ordering of the pages is also important and if a page was visited more than once by a given user in a user defined time interval, then it is relevant as well. For this reason the preprocessing module of the whole system provides the sequences of Web pages by users or user sessions. For sub tree mining not only the sequences are needed

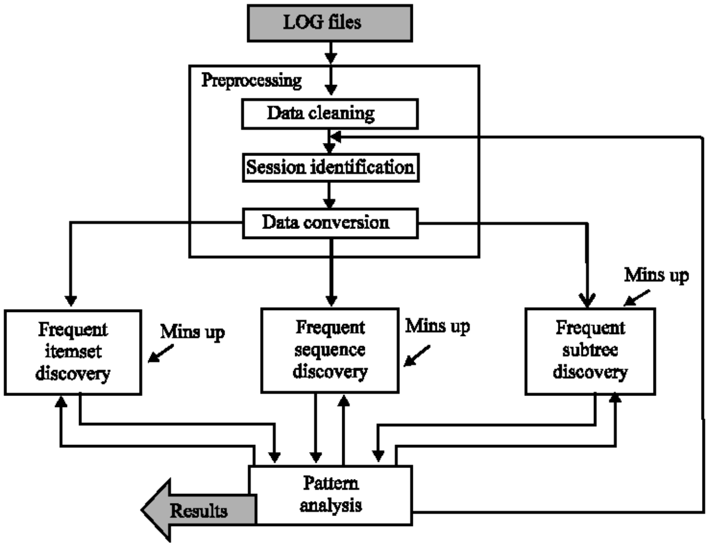


Fig. 1: Process of Web usage mining

but also the structure of the web pages visited by a given user. In this case the backward navigations are omitted; only the forward navigations are relevant, which form a tree for each user. After the discovery has been achieved, the analysis of the patterns follows. The whole mining process is an iterative task which is depicted by the feedback in Fig. 1. Depending on the results of the analysis either the parameters of the preprocessing step can be tuned (i.e. by choosing another time interval to determine the sessions of the users) or only the parameters of the mining algorithms (In this case that means the minimum support threshold).

In the case study presented in this research the aim of Web usage mining is to discover the frequent pages visited at the same time and to discover the page sequences visited by users. The results obtained by the application can be used to form the structure of a portal satisfactorily for advertising reasons and to provide a more personalized Web portal.

In Web usage mining several data mining techniques can be used. Association rules are used in order to discover the pages which are visited together even if they are not directly connected, which can reveal associations between group of users with specific interest. This information can be used for example for restructuring Web sites by adding links between those pages which are visited together. Sequence mining can be used for discover the Web pages which are accessed immediately after another. Using this knowledge the trends of the activity of the users can be determined and predictions to the next visited pages can be calculated. Web usage mining is elaborated in many aspects. Besides applying data mining techniques also other approaches are used for discovering information about navigation patterns. Using PLSA the hidden semantic relationships among users and between users and Web pages can be detected. In Markov assumptions are used as the basis to mine the structure of browsing patterns (Nanopoulos *et al.*, 2002). For Web prefetching, (Nanopoulos *et al.*, 2003; Umapathi and Raja, 2006) uses Web log mining techniques and (Zhang and Ghorbani, 2004) uses a Markov predictor.

THE THREE MINING ALGORITHMS

Before investigating the whole process of Web usage mining and before explaining the important steps of the process, the frequent pattern mining algorithms are explained here briefly. It is necessary to understand the mechanism of these algorithms in order to understand their results. Another important aspect is to determine the input parameters of the algorithm in order to have the opportunity of providing the adequate input formats by the preprocessing phase of the mining process. As mentioned earlier the frequent set of pages are discovered using the ItemsetCode algorithm (Invancsy and Vajk, 2005). It is a level-wise “candidate generate and test” method that is based portionally on the Apriori hypothesis. The aim of the algorithm is to enhance the Apriori algorithm on the low level. It means, enhancing the step of discovering the small frequent itemsets. In such a way also the greater itemsets are discovered more quickly. The idea of the Itemset Code algorithm is to is to reduce the problem of discovering the 3 and 4 frequent itemsets back to the problem of discovering 2 frequent itemsets by using a coding mechanism. The Itemset Code algorithm discovers the 1 and 2 frequent itemsets in the quickest way by directly indexing a matrix. The 2 frequent itemsets are coded and the 3 and 4 candidates are created by pairing the codes. The counters for the 3 and 4 candidates are stored in a jugged array in order to have a storage structure of moderate memory requirements. The way in which the candidates are created enables us to use the jugged array in a very efficient way by using 2 indirections only. Furthermore, the memory requirement of the structure is also low. The algorithm only partially exploits the benefits of the Apriori hypothesis. The reason is the compact storage structure for the candidates. The Itemset Code algorithm discovers the large itemset efficiently because of the quick discovery of the small itemsets. Its level-wise approach ensures the fact that its memory requirement does not depend on the number of transactions. The input format of the Itemset Code algorithm suits the input format of other frequent mining algorithms. It reads the transactions by rows and each row contains the list of items. The page sequences are discovered using the SM-Tree algorithm (State Machine- Tree algorithm) (Invancsy and Vajk, 2005b). The main idea of the SM-Tree algorithm is to test the subsequence inclusion in such a way that the items of the input sequence are processed exactly once. The basis of the new approach is the deterministic finite state machines created for the candidates. By joining the several automatons a new structure called SM-Tree is created such that handling a large number of candidates is faster than in the case of using different state machines for each candidate. Based on its features the SM-Tree structure can be handled efficiently. This can be done by exploiting the benefits of having two types of states, namely the fixed and the temporary states. The further benefit of the suggested algorithm is that its memory requirement is independent from the number of transactions which comes from the level-wise approach. The input format of the SM-Tree algorithms contains rows of transactions, where each row contains a sequence, where the itemsets are separated by a -1 value. The PD-Tree algorithm proposes a new method for determining whether a tree is contained by another tree. This can be done by using a pushdown automaton. In order to provide an input to the automaton, the tree is represented as a string. For handling the large number of candidates efficiently the join operation between the automatons were proposed and the resulting new structure is called PD-Tree. The new structure makes it possible to discover the support of each candidate at the same time by processing the items of a transaction exactly once. The benefit of the PD-Tree is that it uses only one stack to accomplish the mining process.

Experimental results show the time saving when using the PD-Tree instead of using several pushdown automatons. The input format of the algorithm also contains rows of transactions where each transaction contains a tree. A tree is represented with strings.

DATA PREPROCESSING

The data in the log files of the server about the actions of the users can not be used for mining purposes in the form as it is stored. For this reason a preprocessing step must be performed before the pattern discovering phase. The preprocessing step contains 3 separate phases. Firstly, the collected data must be cleaned, which means that graphic and multimedia entries are removed. Secondly, the different sessions belonging to different users should be identified. A session is understood as a group of activities performed by a user when he is navigating through a given site. To identify the sessions from the raw data is a complex step, because the server logs do not always contain all the information needed. There are Web server logs that do not contain enough information to reconstruct the user sessions; in this case for example time-oriented heuristics can be used. After identifying the sessions, the Web page sequences are generated which task belongs to the first step of the preprocessing. The third step is to convert the data into the format needed by the mining Algorithms. If the sessions and the sequences are identified, this step can be accomplished more easily. In our experiments we used two web server log files, the first one was the msnbc.com anonymous data1 and the second one was a Click Stream data downloaded from the ECML/PKDD 2005 Discovery Challenge 2. Both of the log files are in different formats, thus different preprocessing steps were needed.

The msnbc log data describes the page visits of users who visited msnbc.com on july 28, 2007. Visits are recorded at the level of URL category and are recorded in time order. This means that in this case the first phase of the preprocessing step can be omitted. The data comes from Internet Information Server (IIS) logs for msnbc.com. Each row in the dataset corresponds to the page visits of a user within a 24 h period. Each item of a row corresponds to a request of a user for a page. The pages are coded as shown in Table 1. The client-side cached data is not recorded, thus this data contains only the server-side log.

Table 1: Codes for the msnbc.com page categories

Category	Cod	Category	Cod	Category	Cod
Frontpage	1	Misc	7	Summary	13
News	2	Weather	8	Blos	14
Tech	3	Health	9	Travel	15
Local	4	Living	10	msn-news	16
Opinion	5	Business	11	msn-sport	17
On-air	6	Sports	12		

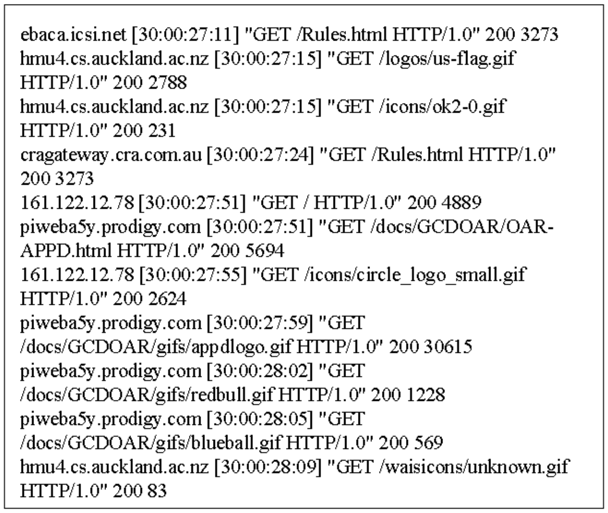
1http://kdd.ics.uci.edu/databases/msnbc/msnbc.html,
2http://lisp.vse.cz/ challenge/CURRENT/

In the case of the msnbc data only the rows have to be converted into itemsets, sequences and trees. The other preprocessing steps are done already. A row is converted into an itemset by omitting the duplicates of the pages and sorting them regarding their codes. In this way the Itemset Code algorithm can be executed easily on the dataset. In order to have sequence patterns the row has to be converted such that they represent sequences. A row corresponds practically to a sequence having only one item in each itemset. Thus converting a row into the sequence format needed by the SM-Tree algorithm means to insert a -1 between each code.

In order to have the opportunity mining tree-like patterns the database has to be converted such that the transactions represent trees. For this reason each row is processed in the following way. The root of the tree is the first item of the row. From the subsequent items a branch is created until an item is reached which was already inserted into the tree. In this case the algorithm inserts as many -1 item into the string representation of the tree as the number of the items is between the new item and the previous occurrence of the same item. The further items form another branch in the tree. For example given the row: “1 2 3 4 2 5” then the tree representation of the row is the following: “1 2 3 4 -1 -1 5”. In case of the Click Stream data, the preprocessing phase needs more work. It contains 546 files where each file contains the information collected during one hour from the activities of the users in a Web store. Each row of the log contains the following parts:

- A shop identifier.
- Time.
- IP address.
- Automatic created unique session identifier.
- Visited page.
- Referrer.

In Fig. 2 a part of the raw log file can be observed. Because in this case the sessions have already been identified in the log file, the Web page sequences for the same sessions have to be collected only in the preprocessing step. This can be done in the different files separately, or through all the log files. After the sequences are discovered, the different web pages are coded and similarly to the msnbc data, the log file has to be converted into itemsets and sequences.



```
ebaca.icsi.net [30:00:27:11] "GET /Rules.html HTTP/1.0" 200 3273
hmu4.cs.auckland.ac.nz [30:00:27:15] "GET /logos/us-flag.gif
HTTP/1.0" 200 2788
hmu4.cs.auckland.ac.nz [30:00:27:15] "GET /icons/ok2-0.gif
HTTP/1.0" 200 231
cragateway.cra.com.au [30:00:27:24] "GET /Rules.html HTTP/1.0"
200 3273
161.122.12.78 [30:00:27:51] "GET / HTTP/1.0" 200 4889
piweba5y.prodigy.com [30:00:27:51] "GET /docs/GCDOAR/OAR-
APPD.html HTTP/1.0" 200 5694
161.122.12.78 [30:00:27:55] "GET /icons/circle_logo_small.gif
HTTP/1.0" 200 2624
piweba5y.prodigy.com [30:00:27:59] "GET
/docs/GCDOAR/gifs/appdlogo.gif HTTP/1.0" 200 30615
piweba5y.prodigy.com [30:00:28:02] "GET
/docs/GCDOAR/gifs/redbull.gif HTTP/1.0" 200 1228
piweba5y.prodigy.com [30:00:28:05] "GET
/docs/GCDOAR/gifs/blueball.gif HTTP/1.0" 200 569
hmu4.cs.auckland.ac.nz [30:00:28:09] "GET /waisicons/unknown.gif
HTTP/1.0" 200 83
```

Fig. 2: An example of raw log file

Here is a sample line of a web log file in its raw format

217.13.12.209 - - [19/May/2001:02:50:32 -0400] "GET /meta_tags.htm HTTP/1.1" 200 28950 "http://www.google.com/search?q=meta+and+tag" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

This web server log file line tells us

Visitor's IP address or hostname [217.13.12.209]
Login [-]
Authuser [-]
Date and time [19/May/2001:02:50:32 -0400]
Request method [GET]
Request path [meta_tags.htm]
Request protocol [HTTP/1.1]
Response status [200]
Response content size [28950]
Referrer path [http://www.google.com/search?q=meta+and+tag]
User agent [Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)]

DATA MINING AND PATTERN ANALYSIS

As it is depicted in Fig. 1, the Web usage mining system is able to use all 3 frequent pattern discovery tasks described in this research. For the mining process, besides the input data, the minimum support threshold value is needed. It is one of the key issues, to which value

Opinion and misc and travel	➔	on-air	90.29%
News and misc and business and bbs	➔	frontpage	90.26%
Living and business and sports and bbs	➔	frontpage	90.25%
News and misc and business and sports	➔	frontpag	90.23%
News and tech and living and business and sports	➔	frontpage	90.21%
News and living and business and bbs	➔	frontpage	89.86%
Frontpage and tech and living and business and sports	➔	news	89.63%
Frontpage and opinion and living and sports	➔	news	89.56%
Frontpage andtech and on-air and business andsports	➔	news	89.44%
News and misc and sports and bbs	➔	news	89.26%
News and tech and on-air and business and sports	➔	frontpage	89.00%
News and living and business and sports	➔	frontpage	88.61%
News and business and sports and bbs	➔	frontpage	87.86%
Misc and living and travel	➔	frontpage	87.81%
Tech andliving and sports and bbs	➔	on-air	87.77%
Tech and business and sports and bbs	➔	frontpage	87.60%
News and misc and living and business	➔	frontpag	87.55%
On-air and business and sports and bbs	➔	frontpage	86.65%
News and tech and misc and bbs	➔	frontpage	86.54%
On-air and misc and business and sports	➔	frontpage	86.48%
Tech and Misc and travel	➔	frontpage	86.16%
Tech and living and business and sports	➔	on-air	85.80%
News and living and sports and bbs	➔	frontpage	85.78%
Misc and business and sports	➔	frontpage	85.69%
Frontpage and tech and opinion and sports	➔	frontpa	85.63%
News and opinion and living and sports	➔	news	85.60%
Misc and business and travel	➔	frontpage	85.25%
News andtech and misc and business	➔	on-air	85.16%

Fig. 3: Association rules on msnbc data

Misc➔ local	2.11%
Frontpage ➔frontpage ➔ sports	2.07%
Local ➔ frontpage	2.01%
On-air ➔frontpage	1.91%
On-air ➔ misc -> on-air	1.87%
On-air ➔ news	1.71%
News ➔ front page ➔ news	1.68%
Local ➔ news	1.51%
Frontpage ➔ front page ➔ business	1.49%
News ➔ sports	1.45%
News ➔ bbs	1.35%
Health ➔ local	1.34%
Misc ➔ frontpage ➔ frontpage	1.31%
On-air ➔ local	1.23%
Misc ➔ on-air ➔ misc	1.21%
frontpage➔ frontpage➔ living	1.19%
local ➔ frontpage➔ frontpage	1.17%
health ➔ misc	1.16%
misc ➔ on-air ➔ on-air	1.16%
local ➔ misc ➔local	1.14%
misc ➔ news	1.13%
news ➔ living	1.13%
on-air ➔ misc ➔ on-air ➔ misc	1.11%

Fig. 4: Sequential rules based on the msnbc data

the support threshold should be set. The right answer can be given only with the user interactions and many iterations until the appropriate values have been found. For this reason, namely, that the interaction of the users is needed in this phase of the mining process, it is advisable executing the frequent pattern discovery algorithm iteratively on a relatively small part of the whole dataset only. Choosing the right size of the sample data, the response time of the application remains small, while the sample data represents the whole data accurately. Setting the minimum support threshold parameter is not a trivial task and it requires a lot of practice and attention on the part of the user. The frequent itemset discovery and the association rule mining was accomplished using the ItemsetCode algorithm with different minimum support and minimum confidence threshold values. Figure 3 depicts the association rules generated from msnbc.com data at a minimum support threshold of 0.1% and at a

summary frontpage bbs	0.15%
health news misc	0.14%
on-air misc – business	0.13%
news sports living	0.12%
frontpage travel – local	0.12%
health local – on-air	0.12%
frontpage tech – news – tech	0.12%
frontpage tech – sports – news	0.11%
frontpage misc – news misc	0.11%
frontpage sports – news – business	0.10%
news travel – on-air misc	0.05%
tech living opinion weather	0.05%
frontpage news – misc – travel – news	0.03%
frontpage news – living – tech – local	0.02%
frontpage tech – living – news – living	0.02%
frontpage sports – news – sports bbs	0.02%

Fig. 5: Frequent tree patterns based on msnbc data in string

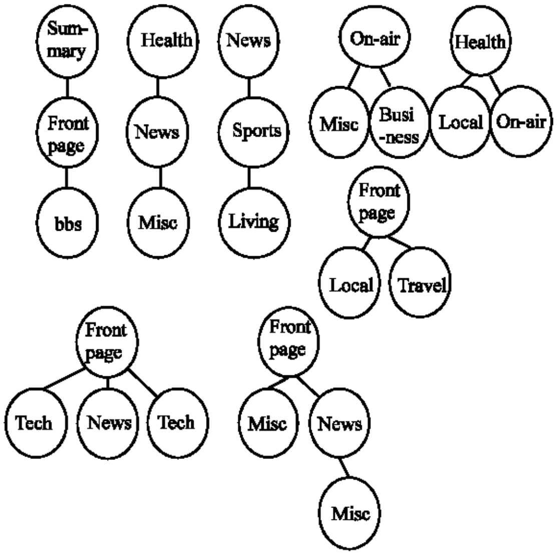


Fig. 6: Frequent tree patterns based on msnbc data in graphical representation

minimum confidence threshold of 85% (which is depicted in the Fig. 3). Analyzing the results, one can make the advertising process more successful and the structure of the portal can be changed such that the pages contained by the rules are accessible from each other. Another type of decision can be made based on the information gained from a sequence mining algorithm. Figure 4 shows a part of the discovered sequences of the SM-Tree algorithm from the msnbc.com data. The percentage values depicted in Fig. 4 are the support of the sequences. The frequent tree mining task was accomplished using the PD-Tree algorithm. A part of the result of the tree mining algorithm is depicted in Fig. 5. The patterns contain beside the trees (represented in string format), also the support values. The graphical representations of the patterns are depicted in Fig. 6 without the support values.

CONCLUSION

This study deals with the problem of discovering hidden information from large amount of Web log data collected by web servers. The contribution of the paper is to introduce the process of web log mining and to show how frequent pattern discovery tasks can be applied on the web log data in order to obtain useful information about the user's navigation behavior.

REFERENCES

Batista, P., M. Ario and J. Silva, 2002. Mining web access logs of an on-line newspaper.

Cooley, R., B. Mobasher and J. Srivastava, 1999. Data preparation for mining world wide web browsing patterns. Knowl. Inform. Sys., 1: 5-32.

Iváncsy, R. and I. Vajk, 2005a. Efficient Sequential Pattern. Mining Algorithms. WSEAS Transactions on Computers, 4: 96-101.

Iváncsy, R. and I. Vajk, 2005b. PD-Tree: A New Approach to Subtree Discovery. WSEAS Trans. Inform. Sci. Applications, 2: 1772-1779.

Kosala and Blockeel, 2000. Web mining research: A survey. SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining, ACM, Vol. 2.

Li Shen, J.F.F.M.V.M., Ling Cheng and T. Steinberg, 2000. Mining the most interesting web access associations. In WebNet 2000-World Conference on the WWW and Internet, pp: 489-494.

Madria, S.K., S.S. Bhowmick, W.K. Ng and E.P. Lim, 1999. Research issues in web data mining. In Data Warehousing and Knowledge Discovery, pp: 303-312.

Nanopoulos, A. and Y. Manolopoulos, 2001. Mining patterns from graph traversals. Data and Knowl. Eng., 37: 243-266.

Nanopoulos, A., D. Katsaros and Y. Manolopoulos, 2000. Exploiting web log mining for web cache enhancement. In: WEBKDD: Springer-Verlag, pp: 68- 87.

Nanopoulos, A., D. Katsaros and Y. Manolopoulos, 2003. A data mining algorithm for generalized web prefetching. IEEE. Trans. Knowl. Data Eng., 15: 1155-1169.

Srivastava, J., R. Cooley, M. Deshpande and P.N. Tan, 2000. Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations, 1: 12-23.

Umapathi, C. and J. Raja, 2006. A prefetching algorithm for Improving Web cache performance. J. Applied Sci., 6: 3122-3127.

Yang, Q. and H.H. Zhang, 2003. Web-log mining for predictive web caching. IEEE. Trans. Knowl. Data Eng., 15: 1050-1053.

Zhang, J. and A.A. Ghorbani, 2004. The reconstruction of user sessions from a server log using improved time oriented heuristics. In: CNSR. IEEE. Comput. Soc., pp: 315-322.