Open access • Proceedings Article • DOI:10.1109/KDEX.1997.629824

Grouping Web page references into transactions for mining World Wide Web browsing patterns — Source link

Robert Cooley, Bamshad Mobasher, Jaideep Srivastava

Institutions: University of Minnesota

Published on: 04 Nov 1997

Topics: Web page, Static web page, Web mining, Web navigation and Web modeling

Technical Report

Department of Computer Science
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 97-021

Grouping Web Page References Into Transactions for Mining World Wide Web Browsing Patterns

by: Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava

1 Introduction and Background

As more organizations rely on the World Wide Web to conduct business, traditional strategies and techniques for market analysis need to be revisited. Organizations collect large volumes of data and analyze it to determine the life time value of customers, cross marketing strategies across products, and effectiveness of promotional campaigns. In the Web, such information is generally gathered automatically by Web servers and collected in server or access logs. Analysis of server access data can provide information on how to restructure a Web site for increased effectiveness, better management of workgroup communication, and analyzing user access patterns to target ads to specific groups of users. Most existing Web analysis tools [Inc96, eSI95, net96] provide very primitive mechanisms for reporting user activity, i.e. it is possible to determine the number of accesses to individual files, the times of visits, and URLs of users. However, these tools usually provide little analysis of data relationships among the accessed files, which is essential to fully utilizing the data gathered in daily transactions. A comprehensive analysis tool must automatically discover such relationships among users accesses.

Several researchers have proposed the application of data mining techniques to facilitate information discovery on global information systems such as the Internet [ZH95, KKS96]. The focus of these proposals is knowledge discovery across the Internet, based on content, and not the analysis of user access patterns on various Web servers. Web server access logs, however, have been used as a testbed for the application of certain data mining tasks such as the discovery of frequent episodes [MTV95]. Recently, *maximal forward references* have been proposed [CPY96] as a way to extract meaningful user access sequences. *Web mining* is the application of data mining techniques to large Web data repositories, examples of which are provided below.

Discovering Association Rules: In of Web mining, an example of an association rule is the correlation among accesses to various files on a server by a given client. For example, using association rule discovery techniques [AS94] we can find the following correlations: (i) 60% of clients who accessed the page with URL `/company/products/`, also accessed the page `/company/products/product1.html`; (ii) 40% of clients who accessed the Web page with URL `/company/products/product1.html`, also accessed `/company/products/product2.html`; and (iii) 30% of clients who accessed
`/company/special-offer.html`, placed an online order in `/company/products/product1`. In Web mining additional properties of data can be used to prune the search space, since information about

a site's structural hierarchy can be used. For example, if the support for `/company/products/` is low, one may conclude that the search for association between the two secondary pages with URLs `/company/products/product1` and `/company1/products/product2` should be pruned since neither are likely to have adequate support.

Discovery of Sequential Patterns: Given a database of time-stamped transactions, the problem of discovering sequential patterns [MTV95, SA96] is to find inter-transaction patterns, i.e. the presence of a set of items followed by another item, in the time-stamp ordered transaction set. In Web server transaction logs, a visit by a client is recorded over a period of time. By analyzing this information, we can determine temporal relationships among data items such as: (i) 30% of clients who visited `/company/products/product1.html`, had done a search in Yahoo, within the past week on keywords w_1 and w_2; and (ii) 60% of clients who placed an online order in `/company/products/product1.html`, also placed an online order in `/company1/products/product4` within 15 days.

As the examples above show, mining for knowledge from web log data has the potential of revealing information of great value. While this certainly is an application of existing data mining algorithms, e.g. discovery of association rules or temporal sequences, the overall task is not one of simply adapting existing algorithms to new data. Because of many unique characteristics of the client-server model in the World Wide Web, including radical differences between the physical and logical data organizations of web repositories, it is necessary to develop a new framework to enable the mining process. Specifically, there are a number of issues in *pre-processing data for mining* that must be addressed before the mining algorithms can be run. These include developing a model of web log data, developing techniques to clean/filter the raw data to eliminate outliers and/or irrelevant items, grouping individual page accesses into semantic units (i.e. transactions), and specializing generic data mining algorithms to take advantage of the specific nature of web log data.

This paper presents a general model for identifying transactions for data mining WWW log data. The specific contributions include (i) definition of generic transaction identification modules, (ii) definition of a user browsing behavior model that can be used to separate important or information *content* page references from references used for *navigation* purposes, (iii) development of specific transaction identification modules based on web page reference lengths or web site structure, (iv) and evaluation of the different transaction identification modules using generated server log data with known association rules.

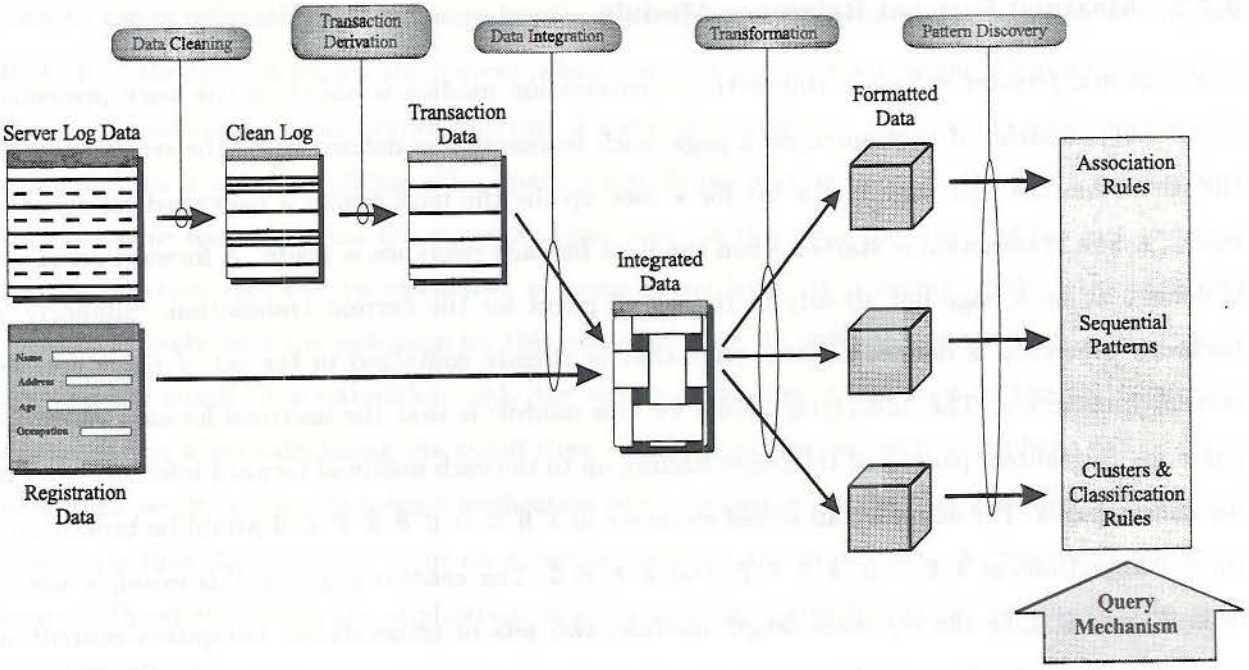

Figure 2: General Architecture for WEBMINER

is unlikely that a fixed time window will break a log up appropriately. However, the *time window* transaction module can also be used as a *merge* module in conjunction with one of the other *divide* modules. For example, after applying the *reference length* module, a *merge time window* module with a 10 minute input parameter could be used to ensure that each transaction has some minimum overall length.

4 The WEBMINER System

The WEBMINER system [MJH+96] divides the web mining process into two main parts. The first part includes the domain dependent processes of transforming the Web data into suitable "transaction" form, and the second part includes the, largely domain independent, application of generic data mining techniques (such as the discovery of association rule and sequential patterns) as part of the system's data mining engine. The overall architecture for the Web mining process is depicted in Figure 2.

Generally, there are a variety of files accessed as a result of a request by a client to view a particular Web page. These include image, sound, and video files; executable cgi files; coordinates of clickable regions in image map files; and HTML files. Thus, the server logs contain many entries that are redundant or irrelevant for the data mining tasks. For example, all the image file entries

are irrelevant or redundant, since as a URL with several image files is selected, the images are transferred to the client machine and these files are recorded in the log file as independent entries. The process of removing redundant or irrelevant entries from the Web server log files is referred to as *data cleaning*. A very simple form of data cleaning is performed by checking the suffix of the URL name. For instance, all the log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG and map are removed from the log.

After the data cleaning, the log entries must be partitioned into logical clusters using one or a series of transaction identification modules as discussed in section 3.

As depicted in Figure 2, access log data may not be the only source of data for the Web mining process. User registration data, for example, is playing an increasingly important role, particularly as more security and privacy conscious client-side applications restrict server access to a variety of information, such as the client IP addresses or user IDs. The data collected through user registration must then be integrated with the access log data. While WEBMINER currently does not incorporate user registration data, various data integration issues are being explored in the context of Web mining. For a study of data integration in databases see [LHS+95].

Once the domain-dependent data transformation phase is completed, the resulting transaction data must be formatted to conform to the data model of the appropriate data mining task. For instance, the format of the data for the association rule discovery task may be different than the format necessary for mining sequential patterns. Finally, a query mechanism will allow the user (analyst) to provide more control over the discovery process by specifying various constraints. The information from the query is used to reduce the scope, and thus the cost of the mining process. The development of general query mechanism along with appropriate Web-based user interfaces and visualization techniques are issues relevant to the future development of the WEBMINER.

5 Creation of Test Server Log Data

In order to compare the performance of the transaction identification modules presented in section 3 for the mining of association rules, a server log with known rules is needed. Mining of association rules from actual web server logs naturally results in different lists of rules for each module, and even for the same module with different input parameters. It was decided to create server logs with generated data for the purpose of comparing the three modules. The model used to create the data is the user browsing behavior model presented in section 2. The data generator takes a file with

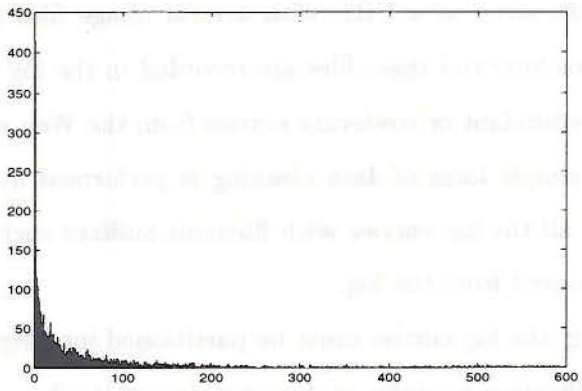

Figure 3: Histogram of Generated Data Reference Lengths (seconds)

a description of a web site as a directed tree or graph and some embedded association rules. The embedded association rules become the "interesting" rules that are checked for during experiments with different transaction identification modules. For each "user", the next log entry is one of three choices, a forward reference, backward reference, or exit. The probability of each choice is taken from the input file, and a random number generator is used to make the decision. If the page reference is going to be a *content* reference, the time is calculated using a normal distribution and a mean time for the page taken from the input file. The times for *navigation* hits are calculated with an exponential distribution. The point of using an exponential distribution for the *navigation* references and a normal distribution with different averages for the *content* references is to create an overall distribution that is similar to those seen in real server logs. Figure 3 shows a histogram of the reference lengths for a generated data set, which is very similar to the histogram of the real server log data shown in Figure 1.

Besides prior knowledge of the "interesting"association rules, the other advantage of using the generated data for testing the transaction identification modules is that the actual percentage of *navigation* references is also known. The obvious disadvantage is that it is after all, only manufactured data and should not be used as the only tool to evaluate the transaction identification modules. Since the data is created using the user behavior model of section 2, it is expected that the two transaction identification modules based on the same model will perform well.

Three different types of web sites were modeled for evaluation, a sparsely connected graph, a densely connected graph, and a graph with a medium amount of connectivity. The sparse graph with an average incoming order of one for the nodes,is shown in figure 4. The medium graph uses the same nodes as the sparse graph, but has more edges added for an average order of four.

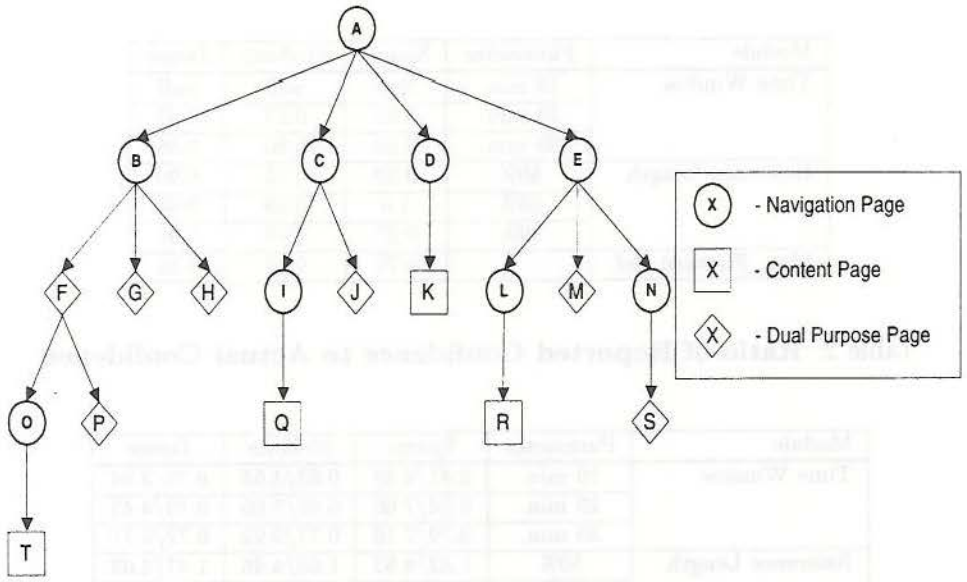

Figure 4: Sparsely Connected Web Site

Module	Parameter	Sparse	Medium	Dense
Time Window	10 min.	0/4	0/3	0/3
	20 min.	2/4	2/3	1/3
	30 min.	2/4	2/3	2/3
Reference Length	50%	4/4	3/3	3/3
	65%	4/4	3/3	3/3
	80%	4/4	3/3	3/3
Max. Forward Ref.		4/4	2/3	1/3

Table 1: **Number of Interesting Rules Discovered**

Similarly, the dense graph has an average incoming edge order of eight.

6 Experimental Evaluation

This section describes the results of using transactions created with the three different transaction identification modules discussed in section 3 to mine for association rules with the WEBMINER system. Section 6.1 details the results of the direct module comparison using the created data and section 6.2 describes association rules found from a real web server log.

6.1 Module Comparison using Created Data

Table 1 shows the results of using the three transactions identification modules to mine for association rules from data created from the three different web site models discussed in section 5. As expected, the *reference length* module performed the best, since it is based on the same model

Module	Parameter	Sparse	Medium	Dense
Time Window	10 min.	null	null	null
	20 min.	0.82	0.87	0.87
	30 min.	0.98	0.90	0.88
Reference Length	50%	0.99	0.95	0.96
	65%	1.0	0.99	0.96
	80%	0.97	0.99	0.96
Max. Forward Ref.		0.79	0.47	0.44

Table 2: **Ratio of Reported Confidence to Actual Confidence**

Module	Parameter	Sparse	Medium	Dense
Time Window	10 min.	0.81/4.38	0.82/4.65	0.75/3.94
	20 min.	0.84/7.06	0.80/7.06	0.73/4.42
	30 min.	0.79/7.16	0.77/9.95	0.72/5.17
Reference Length	50%	1.82/4.62	1.66/4.46	1.47/4.09
	65%	1.68/4.29	1.72/4.35	1.45/4.02
	80%	1.62/4.14	1.66/4.26	1.48/4.03
Max. Forward Ref.		1.26/3.98	1.30/3.95	1.20/3.87

Table 3: **Module Run Time (sec) / Total Run Time (sec)**

that is used to create the data. The *maximal forward reference* module performs well for the sparse data, but as the connectivity of the graph increases, its performance degrades. This is because as more forward paths become available, a *content* reference is less likely to be the "maximal forward reference." For dense graphs, *navigation-content* transactions would probably give better results with the *maximal forward reference* module. The performance of the *time window* module is relatively poor, but as the time window increases, so does the performance.

Table 2 shows the average ratio of reported confidence to actual confidence for the interesting rules discovered. The differences between the *reference length* and *maximal forward reference* modules stand out in table 2. The reported confidence of rules discovered by the *reference length* module are consistently close to the actual values. Note that even though the created data has an actual *navigation* page ratio of 70%, inputs of 50% and 80% produce reasonable results. The reported confidence for the rules discovered by the *maximal forward reference* module is significantly lower than the actual confidence, and similar to the results of table 1, degrades as the connectivity of the graph increases. Table 3 shows the running time of each transaction identification module, and the total run time of the data mining process. The total run times do not include data cleaning since the data was generated in a clean format. Although the data cleaning step for real data can comprise a significant portion of the total run time, it generally only needs to be performed once

Module Used	Confidence(%)	Support(%)	Association Rules
Reference Length (content-only)	61.54	0.18	/mti/clinres.htm /mti/new.htm ⟹ /mti/prodinfo.htm
Reference Length (content-only)	100.00	0.15	/mti/Q&A.htm /mti/prodinfo.htm /mti/pubs.htm ⟹ /mti/clinres.htm
Reference Length (content-only)	26.09	0.14	/cyprus-online/dailynews.htm ⟹ /mti/Q&A.htm
Maximal Forward Reference (content-only)	52.17	0.14	/cyprus-online/Magazines.htm /cyprus-online/Radio.htm ⟹ /cyprus-online/News.htm
Maximal Forward Reference (nav-content)	73.50	1.32	/mti/clinres.htm /mti/new.htm ⟹ /mti/prodinfo.htm

Table 4: **Examples of Association Rules from www.global-reach.com**

for a given set of data. The *time window* module shows the fastest module run time, but a much slower overall data mining process due to the number of rules discovered. The *reference length* module has the slowest module times due to an extra set of file read/writes in order to calculate the cutoff time. All three of the modules are O(n) algorithms and are therefore linearly scalable.

6.2 Association Rules from Real Data

Transactions identified with the *reference length* and *maximal forward reference* modules were used to mine for association rules from a GRIP. The server log used contained 20.3 Mb of raw data, which when cleaned corresponded to about 51.7K references. Because the GRIP web server hosts web sites for other companies, the server log is really a collection of smaller server logs and the overall support for most discovered association rules is low. Accordingly, the association rule generation algorithm was run with thresholds of 0.1% support and 20% confidence. This led to a fairly large number of computed rules (1150 for the *reference length* module and 398 for the *maximal forward reference* module). Table 4 shows some examples of association rules discovered.

The first two rules shown in table 4 are straight forward association rules that could have been predicted by looking at the structure of the web site. However the third rule shows an unexpected association between a page of the *cyprus-online* site and a page from the *MTI* site. Approximately one fourth of the users visiting `/cyprus-online/dailynews.htm` also chose to visit `/mti/Q&A.htm`. Since the *cyprus-online* page no longer exists on the GRIP server, it is not clear if the association is the result of an advertisement,a link to the *MTI* site, or some other factor. The fourth rule listed in table 4 is one of the 150 rules that the *maximal forward reference* module discovered that was not

discovered by the *reference length* module. While the *reference length* module discovered many rules involving the *MTI* web site, the *maximal forward reference* module discovered relatively few rules involving the *MTI* site. An inspection of the *MTI* site revealed that the site is a fully connected graph. Consistent with the results of section 6.1, the *maximal forward reference* module does not perform well under these conditions. The association rule algorithm was run with *navigation-content* transactions created from the *maximal forward reference* module to confirm the theory that the rules missed by the *content-only* transactions would be discovered. The last rule listed in table 4 is the same as the first rule listed, and shows that the *navigation-content* transactions from the *maximal forward reference* module can discover rules in a highly connected graph. However, at thresholds of 0.1% support and 20% confidence, approximately 25,000 other rules were also discovered with the *navigation-content* module.

7 Conclusions

This paper has presented a general model for transaction identification for Web mining, the application of data mining and knowledge discovery techniques to WWW server access logs. This paper also described WEBMINER, a system based on the proposed framework, and presented experimental results on created data for the purpose of comparing transaction identification modules, and on real-world industrial data to illustrate some of its applications (Section 6).

The transactions created with the *reference length* module performed consistently well on both the real data and the created data. For the real data, only the *reference length* transactions discovered rules that could not be reasonably inferred from the structure of the web sites. Since the important page in a traversal path is not always the last, the *content-only* transactions identified with the *maximal forward reference* module did not work well with real data that had a high degree of connectivity. The *navigation-content* transactions led to an overwhelmingly large set of rules, which limits the value of the data mining process. Future work will include further tests to verify the user browsing behavior model discussed in section 2 and tests with transactions created from combinations of *merge* and *divide* modules.

Currently, the implementation of WEBMINER is being extended to incorporate mechanisms for clustering analysis and discovery of classification rules. The query mechanism in WEBMINER will also be extended to include clustering and classification constraints. Also an important area of ongoing research is to continue to develop methods of clustering log entries into user transactions,

including using criteria such as time differential among entries, time spent on a page relative to the page size, and user profile information collected during user registration.

References

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*, pages 487–499, Santiago, Chile, 1994.

[CPY96] M.S. Chen, J.S. Park, and P.S. Yu. Data mining for path traversal patterns in a web environment. In *Proceedings of the 16th International Conference on Distrib uted Computing Systems*, pages 385–392, 1996.

[eSI95] e.g. Software Inc. Webtrends. *http://www.webtrends.com*, 1995.

[Inc96] Open Market Inc. Open market web reporter. *http://www.openmarket.com*, 1996.

[KKS96] I. Khosla, B. Kuhn, and N. Soparkar. Database search using informatiuon mining. In *Proc. of 1996 ACM-SIGMOD Int. Conf. on Management of Data*, Montreal, Quebec, 1996.

[LHS+95] E. Lim, S.Y. Hwang, J. Srivastava, D. Clements, and M. Ganesh. Myriad: Design and implementaion of federated database prototype. *Software - Practive & Experience*, 25(5):533–562, 1995.

[Luo95] A. Luotonen. The common log file format. *http://www.w3.org/pub/WWW/*, 1995.

[MTV95] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pages 210–215, Montreal, Quebec, 1995.

[MJH+96] B. Mobasher, N. Jain, E. Han, and J. Srivastava. Web Mining: Pattern Discovery from World Wide Web Transactions. Technical Report TR 96-050, Dept. of Computer Science, University of Minnesota, Minneapolis, 1996.

[net96] net.Genesis. net.analysis desktop. *http://www.netgen.com*, 1996.

[Pit97] J. Pitkow. In Search of Reliable Usage Data on the WWW. In *Proc. of the Sixth International World Wide Web Conference*, pages 451–463, Santa Clara, CA, 1997

[SA96] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proc. of the Fifth Int'l Conference on Extending Database Technology*, Avignon, France, 1996.

[ZH95] O. R. Zaane and J. Han. Resource and knowledge discovery in global information systems: A preliminary design and experiment. In *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pages 331–336, Montreal, Quebec, 1995.