Users Interest Correlation through Web Log Mining

F. Tao, P. Contreras, B. Pauer, T. Taskaya and F. Murtagh

School of Computer Science, the Queen's University of Belfast; DIW-Berlin

Abstract

When more and more people use web-based information, information of how they use the information is also available in the form of log data. Analysing such data can help information provider to understand their clients' interests over the information space being served, and adapt it according to users point of view. This paper describes a novel way of applying data mining techniques on Internet logging data in order to find correlated web sections from users' point of view. We explain how data from the log file can be transformed into a set of transactional click-streams and how data mining techniques can be applied on these transactions. A test bed has been developed for transforming web log data and discovering association rules from it. Real log data from Microsoft web site is used in experiments and evaluation results show that the approach is effective in obtaining useful knowledge of users correlated interests at a particular web site. We also make some effort on mining other service log data obtained from IRAIA project, an information retrieval system serving economical data.

1. Objectives:

The main objective of web log mining described in this paper is to extract interesting and potentially useful patterns that show users correlated preferences in accesses to the web pages being served by a particular web server. We investigated various methods on web log mining, most of them provide very primitive mechanisms for reporting statistical fact of the accesses, i.e., the number of the accesses to individual files during a period of time, the originality of the users, etc. We believe that by using data mining techniques and systematically analysing the behaviour of past visitors, more sophisticated knowledge of the users access pattern can be obtained from the web log file. In this paper, we discuss how data in the web log file can be transformed to transactional data and fed to an adapted association rules mining algorithm for correlated pages from users point of view. There are several kinds of potential usage on this kind of knowledge obtained: From the site maintaining point of view, it is important reference knowledge for the site administrator to maintain a reasonable and well-orientated web layout. Usage information can also be used to directly aide site navigation by hinting a list of "popular" correlated pages from the current page, thus suggesting new users with the "experience" accumulated from past visitors.

Since the data in the log file does not come ready for mining, we need a data preparation phase where a transaction of user sessions is extracted from the raw web log file. Specially, there are a number of difficulties in cleaning the raw web log file to eliminate irrelevant items, reliably identifying users and user sessions. In this paper, we use a customised transactional model and an algorithm to convert raw web log data into user sessions. Besides this, our contributions include 1. Development of a data mining test bed with association rules mining currently implemented. 2. Experiments carried out in this test bed and evaluation on real web log data shows that the approach is effective in revealing interesting correlated interests from visitors point of view.

The rest of the paper is organised as follows: section 2 reviews related work of analysing web log data. Section 3 explains the model and algorithm used for user session extraction in the data preparation phase. In section 4, we describe the association rule mining in the context in web log mining and experiment results and evaluations of real web log data are presented in section 5. Finally in section 6, we make the conclusion and also describe some initial efforts at adapting this approach to mine users behaviour at an economical information retrieval system, where a customised logging mechanism is designed to collect service usage.

2. Related work

Researches on knowledge discovery from web log data was started originally from web server log analysis tools that provide useful information about user activities and statistical fact of the pages being visited from individual users. The results can provide knowledge such as the number of accesses to individual files during a period of time, the originality of the users, etc. [S01] is a very successful case and more cases are review there. However, these approaches

are usually limited in the ability of providing data relationships among the pages, which is often essential when studying visitors correlated interests at the web site.

The idea of applying data mining techniques to web log data was first proposed in [ZXH98], etc., where the authors investigated the probabilities of applying techniques of clustering, predicating and cycle detecting etc. In [TM00], attempts was made by proposing a framework of web log mining, where log data is modelled as transactions and interest context rule is targeted for mining. Chen, etc. in [CPY96] introduce the concept of using maximal forward reference in order to break down user sessions into transactions for the mining of traversal patterns. A maximal forward reference is the last page requested by a user before backtracking occurs, where the user requests a page previously viewed during that particular user session. Another system described in [YJMD96] uses the knowledge of users access patterns to dynamically maintain the linkage of the web content.

In [IRAIA], a novel Information Retrieval prototype is designed. Information space related to "Economy" can be grouped according to categories like regions, industries, and variables and stored in concept hierarchies. Users search or navigate the information space by selecting entries in the concept hierarchy tree for each of the categories. Therefore, the log data of users interaction with the concept hierarchies and how the entries that they select from the categories to form the queries can contain useful information on users query patterns when using the IR service. The logging mechanism has been implemented by Bernd Pauer and currently being tested by Pedro, we hope enough log data will be collected soon for testing.

3. Data Preparation

There are many reasons why we have to give the raw log a data preparation before being able to apply data mining techniques on it. The most essential ones are the incompatibility of data structure and irrelevant information concerning to the specific mining task. Therefore, the basic work is to transform the data into data-mining friendly form and filter out irrelevant information.

By following HTTP protocol, visitors (by means of browser client) communicate with a web server. Web log file is a plain text (ASCII) file maintained by the server's logging daemon at server side. Each HTTP protocol transaction, whether completed or not, is recorded in the log file. Figure 3.1 shows a single HTTP access that is recorded in the log file. A brief description of the fields is list below. For details, please refer to HTTP protocol detailed in CERN and NCSA [L97]

```
dejh.ipm.ac.ir - - [08/May/2000:00:47:07 -0700]
"GET /spires/form/hepfnal.html HTTP/1.0" 200
3529
"http://www-
spires.slac.stanford.edu/spires/forms.html"
"Mozilla/4.05 [en] (Win95; I)"
```

Figure 3.1 A single HTTP transaction logged at the log file

***Client*:** visitor's domain name or IP address that can be resolved to domain name
Auth*: username if registered
Timestamp: Date and time of the access
Request: request method, document path and name, parameters, etc.
Status: status code indicating the result of a request
Cookie*: Crookie ID
Referrer*: previews link address
User agent*: client side browser type

Figure 3.2 HTTP Server's Log Structure
(*:- optional fields)

Unlike the classical basket data mining solutions [AS94] where transaction is defined as a list of itemsets, there is no natural section of a user transaction in the web log data. For some reason, the web log mechanism simply records every HTTP request when they come from the clients side. And even sometimes, a single request may trigger other file requests. This along with the HTTP server's multi-threads and multi-users features makes users navigational traces nest together in the raw log data. A semantic transaction must be defined with extra effort to meet the demand of mining visitors correlated interests inside a web site, i.e., adapting them to the association rule mining framework. For this purpose, we define a transaction model with capacity to adjust to session number smoothly according to a time-window parameter. We also define the concept of sessions and discuss how to form transactions from the log entries.

Definition 3.1 Let $e \in E$, E is the set of all log entries after data cleaning and e is a log entry object with direct attributes such as host, document, timestamp etc, and derived features such as dwell time, interestingness and session ID, etc. (for example, e.interestingness, interestingness$\in$D.) We denote all these attributes as D and the Domain of D as Dom(D).
A Transaction E[n] can be defined in the following model:

E[n] = { E[n][s] | E[n][s].host = E[n].host AND E[n][s+1].timestamp - E[n][s].timestamp < W.ΔT, 1≤ s ≤|E[n]| }

where n is the transaction ID number, s is the session ID number within a specific transaction, ΔT is the time interval configured in the window object W, which will be defined in Definition 2.2. The following illustrates these transaction and session concepts. We can notice that all the log data have been transformed to transactions and sessions, the session is an object with various encapsulated measurements contributing to the user's overall behavior. In another word, the transaction can be regarded as a click-stream generated by a single visitor within a specific time window.

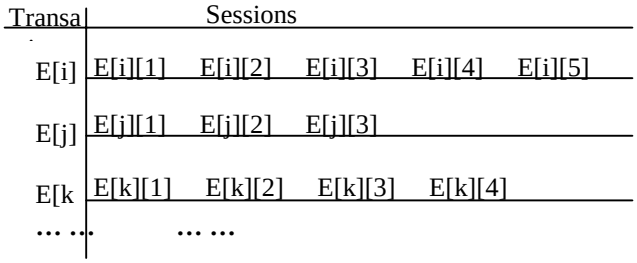

Figure 3.3 Transactions and user sessions

Definition 3.2 A time window W in a transaction set E is a mechanism to categorize sessions into different transactions according to its integrated attribute window-size ΔT, which can be adjusted to simulate the time-span of the browsing transaction semantic, we set it as 20 minutes in our case.

Definition 3.3 Session s = E[n][i] is the ith element in a specific transaction E[n], where $1 \le i \le |E[n]|$, s is an object encapsulating all the attributes and features that are possibly derived from either a single log entry or the context entries within a single transaction.

Implementation of the data transformation
By using the model, We transform the raw log data into a customised database that follows the data format needed in association rule mining. We will explain the association rule mining in next section. Each entry in the log file can be regarded as a vote to the web site and a section name is part of the vote to implicate the visitor's preference to that section. The raw log data is scanned for twice: in the first scan, sections, each of which can be a page or a group of pages in a category, are indexed and assigned with IDs. In this step, a filtering and counting mechanism are also involved in. Data is read from the raw log file line by line and then fed into a customised filter where irrelevant entries such as icons and auxiliary graphic etc. can be ignored. These requests are often automatically triggered when requesting a content-based page, therefore can not be counted to measure users visiting patterns. For the rest of the data that comes out of the filter, we save them in a hash table where each of the hash entry is a (key, value) pair. We use the section as the key and the value is an object keeping the section ID and maintaining the section count. At the end of the first scan, irrelevant log entries are ruled out and each of the rest of them is assigned with a distinctive ID. We use hash table to speed up the second scan over the log data where a name and ID correspondence can be easily located.

In the second scan, we maintain a bookmark array to store positions of the file pointers to log entries of the same transaction, i.e., the client addresses are the same and the visiting timestamps fall into the same time window. For each of the log entry read from the log file, if it is indexed in the hash table, the file pointer is not already recorded in the bookmark array and it also belongs to a same transaction, then we model it as a new session and look up the hash table and use the ID to identify that session. When we finish the whole time window, we get a transaction of sessions represented by section IDs. More features can be obtained and stored for other customised mining tasks, interested reader can refer to [TM00] for further reading.

4. Justification of Association Rule Mining for Correlated Interests Mining

The association rule mining task was first introduced in [AS94] for mining relationships in large database. In the context of correlated interests mining from web log file, it can be described as follows: let S be the site space represented by a set of pages or sections. And T the transactions, where each transaction represents a click stream made by a single visitor within a specific time window. A transaction is a set of items representing all the pages requested in a single visit. A set of k items is called k-itemset. The support of an itemset X, denoted support(X), is the number of transactions in which it occurs as a subset. An itemset is frequent if its support is more than a user-specified minimum support (min_sup) value.

An association rule is in the form of A => B, where A and B are itemset. The support of the rules is given as support(A∪B), and the confidence as support(A∪B)/support(A), i.e., the conditional probability that a transaction contains B, given it contains A. We say A and B are strong correlated if the confidence is above the pre-specified minimum confidence.

The data-mining task is to generate all association rules in the database, which have a support greater than min_sup and a confidence greater than min_conf. We give the algorithm in the context of correlated interests mining from web log data.

1) L_1={large 1-itemsets};
2) **For** (k=2;$L_{k-1}\neq 0$;k++) **do begin**
3) C_k=Apriori-Gen(L_{k-1});
4) **forall** transactions TID $\in$ D **do begin**
5) C_t = D[TID];
6) **forall** candidates c $\in C_t$ **do**
7) c.count++;
8) **end**.
9) L_k={c $\in C_k$ | c.count $\geq$min_sup}
10) **end**.
11) Results $\cup_k L_k$;

C_k : set of candidate k-itemsets, the items of which are potentially frequent. A support count is associated with each of them;
L_k: Set of frequent k-itemsets, which is a subset of C_k and have all the support value of its items above the min_sup. L_1 is constructed by collecting frequent sections from the hash table obtained in the data preparation phase, i.e., the count of the section is above min_sup.
D: the dataset, which is a series of transactions indexed by TID (Transaction ID)

Apriori-Gen – Candidate Generation: This sub-Function takes the set L_{i-1} as the input parameter, through self-joining of the items in L_{i-1}, it returns a power set of L_{i-1}, denoted as C_i, which includes all the potential frequent candidates itemsets. Two frequent itemsets of size i-1 with the same items in their first i-2 items joined, generating a new candidate itemset of size *i*:

Insert into C_i
Select $p.item_1$, $p.item_2$, …, $p.item_{i-1}$, $q.item_{i-1}$
From L_{i-1} p, L_{i-1} q
Where $p.item_1=q.item_1$, …, $p.item_{i-2}=q.item_{i-2}$, $p.item_{i-1}<q.item_{i-1}$;

Then, the candidate set C_i is produced by removing every candidate *i*-itemset c such that some (i-1)-subset of c is not in L_{i-i}:
Forall candidate itemsets c $\in C_i$ **do begin**
Forall (i-1)-subsets s of c **do begin**
If (s $\notin L_{i-1}$) then
Delete c from C_i;

In the algorithm above, the first iteration computes the set L_1 of frequent 1-itemsets. A subsequent iteration *i* consists of two phases. First, a set C_i of candidate *i*-itemset is created by joining the frequent (*i*-1)-itemsets in L_{i-1} found in the previous iteration. This phase is realized by the Apriori-Gen sub-function described below. Next, the database is scanned to determine the support of the candidates in C*i* and the non-frequent *i*-itemsets are ruled out of the candidate set. This process is repeated until no more candidates can be generated.

5. Experiment on web log mining and evaluation result

The approach explained above is used to mine for correlated interests from Microsoft web log data provided by UCI Knowledge Discovery Dataset Archive [UCIDA]. The data is transformed and any information that could identify a specific visitor has been pre-processed. The data records the use of www.microsoft.com by 32710 anonymous transactions. 294 web sections are involved in the log file after filtering.

We implement the transaction model and the mining algorithm in Java and the following screenshot shows the mining result with threshold of 2% minimum support and 70% minimum confidence.

Figure 5.1 correlated interests discovered at threshold of (2%, 70%) as min sup and min conf

The mining result shows that visitors of Microsoft web site have many correlated interests. After sorting the rules in descending support order, we found that the most frequent correlated interests are “Window 95 support” and “isapi”. “isapi”, after being looked into the site, proved to be one of the Microsoft’s main portal URL forwarder where requests are forwarded to the right sections indicated in the URL. Obviously, this rule indicates the knowledge that “Window95 support” takes the largest portion from the requests that were sent to the portal. The second most frequent rule is “Knowledge Base, isapi” => “Support

Desktop". This shows that the Desktop support is the high light in Microsoft Knowledge base from users point of view. On the other side, if we sort the rules according to their confidences, we get a list of rules ordered by the their correlation strengths. The most strong correlation is between "Windows 95" and Windows Family of OS" (in the third line). This indicates when "Windows 95" is visited, it is very likely (with probability of 91.5%) that the Windows Family of OSs is visited as well in that single transaction.

By adjusting the minimum support and confidence, we can discover more correlated interests with low support but high confidence or reverse. Obviously this knowledge can be essential to understand visitors interests correlation and access pattern in the web site.

6. Summary and future work

In this paper, we investigated the possibility of analysing web log data by using data mining techniques, more specifically, association rule mining for correlated interests patterns that web site visitors left in the web log data. We described a data preparation phase where raw data in the web log file is transformed into transaction forms using a transaction model. We then described the association rule mining model and algorithm in the context of mining for correlated interests. We implemented a test bed and evaluated the approach on real log data obtained from Microsoft web site. Results show that the approach is effective in discovering visitors correlated interests when browsing a specific web site.

As described in section of related work, we have started the research on integrating a logging mechanism in the IRAIA economic information retrieval system. The logging mechanism will record all the database queries that the users generate by interacting with a GUI based query interface. Similar approach is expected to be applied on the log data and mine for correlated interests. result rules are stored in a knowledge base, which will be used by a hinting mechanism to provide new users with the "experience" of the past users and help them to interact with the query interface of the IR system.

References:

[AS94] R. Agrawal and R. Srikant: *Fast Algorithms for Mining Association Rules.* Int. Conference of Very Large Data Bases, Santiago, Chile, September 1994. pp 487-499

[CPY96] M.S. Chen, J.S. Park, P.S.Yu: *Data Mining for Path Traversal Patterns in a Web Environment.* 16th Intl. Conference on Distributed Computing System, 1996. pp385-392

[YJMD96] T.Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal: *from user access patterns to dynamic hypertest linking.* Intl World Wide Web Conference, Paris, France, 1996

[TM00] F. Tao, F. Murtagh: *Towards Knowledge Discovery from WWW Log Data.* Intl. Conference on Information Technology: Coding and Computing. Mar. 2000. pp 302-307

[ZXH98] O.R. Zaïane, M. Xin, J.W. Han: *Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs.* ADL 1998. pp19-29

[CT01] P. Contreras, F. Tao: *The Integration of the log mining in the context of IRAIA.* Presented at DIW technical meeting, Berlin, Feb. 2001. http://iraia.diw.de/

[S01] S. Turner. Analog – A Web Log Analyst. http://www.statslab.cam.ac.uk/~sret1/analog/

[L97] A. Luotonen. The common log file format. *http://www.w3.org/pub/WWW/*, 1997

[IRAIA] Getting Orientation in Complex Information Spaces as an Emergent Behaviour of Autonomous Information Agents. http://iraia.diw.de/