Click-stream Data Analysis of Web Traffic

Riivo Kikas, Karl Potisepp

December 25, 2009

Abstract

In this paper, we conduct a case study of mining frequent user access patterns from web log files. Our primary objective was to discover the most frequent browsing patterns by analyzing the visitors’ browsing sessions. The nature of pattern mining carried out was mainly exploratory, concentrating on frequent item sets and sequence mining. To achieve this, we eliminated irrelevant data from the access logs, extracted user sessions and searched them interesting usage patterns. During this work we experimented with several frequent itemset mining algorithms and tools. As a result of our work we provide this case study and some statistics about the usage patterns of the public web site of the Faculty of Mathematics and Computer Science of the University of Tartu.

Contents

1 Introduction 2
2 Data preparation 2
  2.1 Data cleaning ............................................. 3
  2.2 Session identification ..................................... 3
  2.3 Statistics about the logs ................................... 4
3 Frequent item sets 5
4 Frequent sequential patterns 7
  4.1 Page access times ........................................ 10
5 Maximal reference sequences 10
6 Changes in patterns through time 10
7 Restricting patterns by user specified pages 12
8 Summary 12
1 Introduction

Understanding how users navigate and browse through web sites is important in many aspects, such as content recommendation, personalization and targeted advertising. Even more, it helps to understand users’ needs and provides real world input to better organization of content and structure. Web server log files can provide important insight to the users’ behavior, because from this data one can reconstruct a reasonably accurate overview of how users navigate their way to whatever content they desire. Coupled with data mining algorithms it is therefore quite probable that important knowledge can be gained.

As an experimental case study, we studied the web access logs from the public web site of the Faculty of Mathematics and Computer Science of the University of Tartu\[1\]. Our aim was to try and find meaningful use-cases from log files and identify possible bottlenecks in the presentation layer and provide some input for user interface redesign.

2 Data preparation

For each time an user visits a web page, a piece of information is typically stored in a web server’s log file. When user navigates around a website, all such pieces of information (alternatively called requests or clicks) form a session (click-stream). We consider user’s single purposed visit (i.e. a search for a phone number or information about entrance exams) as a session and associate each user click with a particular session.

Input data is stored in website log files, namely in Common Log Format\[Wiki09\] format. Each log entry consists of: (i) User’s IP address, (ii) Access time, (iii) Request method (GET, POST), etc), (iv) URL of the page accessed, (v) Protocol (typically HTTP/1.0), (vi) Return code, (vii) Number of bytes transmitted (Figure 2).

Log files contain all HTTP requests made to the web server, including requests for downloading images and style sheet documents attached to the web pages served.

Two important tasks must be performed before click-stream data from CLF files can be used for analyzing: data cleaning and session identification.
2.1 Data cleaning

Since log files contain all requests made to the server, we need to extract only relevant requests and eliminate others. When filtering the requests from log files, we applied the following rules:

- Only requests to the public site are considered. Requests to personal home pages (URLs starting with ) are discarded. Also, all static content files that were used by web pages were removed, such as image files, JavaScript and CSS documents. The server also had a special error page URL that the user was redirected to when they tried to access a non-existent resource. We ignored this kind of access requests, as they are not initially made by users. It must be noted that the request for the original page that caused the server to redirect the user remains in the log file.

- All other HTTP request types besides POST and GET were ignored. We noticed that there were regular OPTIONS requests, that most probably were made by some automated monitoring tool.

- We composed a blacklist of IP addresses, whose requests were all ignored. An IP-address was added to blacklist if there was a request made from the IP address to access robots.txt file. Namely, we believe that only automated indexers request robot.txt and therefore we can ignore all subsequent requests from those IP-s, since we are only interested in how a human user perceives the web site.

Moreover, we modified requested URLs by removing query string attributes (the part of the URL that starts with ?). We assumed that typically URLs identify a distinct page and query-string some operation or action on it. This helped us to reduce the number of different URLs while preserving meaningful information about the usage patterns.

After performing all the procedures described above, the log files contained only information about the HTTP requests originating from intentional clicks made by the users on pages. These clicks can be grouped into users’ click-streams.

2.2 Session identification

The next step was to group user requests made during one visit into sessions. A user session is defined as a sequence of temporally compact accesses by a user [JK00].

Generally, this can be done by providing a cookie to each user with some generated unique identification number and then logging each request and the cookie ID simultaneously. However since CLF does not store an explicit
session ID, and since we could not modify the existing software stack, we had to reconstruct user sessions from the log files.

As a solution, we used a timeout based method to group requests into sessions. We consider a sequence of requests from the same IP address ordered by time to be in same session if the time between consecutive requests is no more than $t = 30$ minutes. If it exceeds 30 minutes, we consider these requests to be part of a new session. While this method is probably not the best, we consider it sufficient as it mimics the way session timeouts are measured on many web sites. We also consider it reasonable to assume that if an user has not interacted with the server for more than $t$ minutes, any new interaction from an user from the same IP address can be thought of as a new use case scenario.

2.3 Statistics about the logs

The log files contained info about requests made to the server from August to December 2009, with October data missing. In total, log files contained 544237 requests, of which 122822 remained after eliminating uninformative lines. These requests were grouped to 35952 sessions. The average session length was 3.4 clicks. Distribution of user session length can be seen on figure 2. Requests involved 2059 distinct URLs.

![Session length histogram](image)

Figure 2: Session length histogram with session length in minutes.
3 Frequent item sets

From the perspective of designing a web site's navigation system, a very important question is whether or not there are groups of pages users tend to visit together. The presence of such groups can give a web designer valuable insight into how people use the web site. For example, content split on separate pages but still often viewed together can be merged to achieve a smoother browsing experience and pages which are visited only to reach other more important (to the user) pages can be eliminated.

We approached the task of identifying such groups from the perspective of frequent item set mining by [AAP01]. We considered every page as a unique item and each session as a transaction or more accurately, an item set. To find pages that are frequently visited together, we searched among these sets for subsets of pages that frequently appear together. The problem now was deciding when a subset of pages is frequent. We solved this by using a frequent item set mining concept called support.

Support is a number that shows how frequently a subset appears in all item sets. Using this concept, it is trivial to eliminate subsets that are not common enough (and therefore not interesting). The only problem is that there is no magic bullet solution to choosing the 'right' support threshold. A value too high does not provide insight into more specific subsets, a value too low results in too much non-frequent garbage.

![Figure 3: The number of frequent closed itemsets changes as the support threshold increases.](image)

However even if we manage to choose an optimal value for the support threshold, after filtering out all the infrequent subsets, some redundancy remains in the results. It is easy to see that if a set of 5 pages appears frequently throughout the sessions, all 4 page subsets of that set appear at least as frequently. To eliminate this kind of noise, we used the Apriori
algorithm [AIS93] to first find all frequent item sets, from which we then extracted *closed itemsets*. In addition to the difficulties in choosing an ideal support threshold, there are other problems with this approach to detecting pages frequently requested during one session. While we can easily find groups of pages that are probably viewed together, much more insight could be gained if we knew the order in which the pages were navigated. By knowing this we would also have an idea of how the users find what they need on the site. Similarly, if an user visits a page several times during one session, this fact is not known to us. Using *sequential item sets* solves these problems.

However, even ordinary frequent item site mining can provide some information. From figure 6, it can be seen that the root page of the web site (*avaleht*) is one of the most frequently accessed resources, which is expected. The advantage of frequent pattern mining can be explained with the itemset where pages *avaleht, 15208, inimesedInstituudid* are present together: it shows that navigation usually starts from the root page and also that the page containing links to different departments’ contact information (*inimesedInstituudid*) is used as a gateway to the page listing all the faculty members (*15208*). The order in which the pages were navigated is not obvious, and looking at the web site we see that there do exist links to go either way. We provide a solution to this problem in the next section.

<table>
<thead>
<tr>
<th>Support</th>
<th>frequent itemset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63</td>
<td>avaleht</td>
</tr>
<tr>
<td>0.13</td>
<td>15213</td>
</tr>
<tr>
<td>0.1</td>
<td>inimesedtudengid</td>
</tr>
<tr>
<td>0.1</td>
<td>avaleht, 15213</td>
</tr>
<tr>
<td>0.09</td>
<td>inimesedInstituudid</td>
</tr>
<tr>
<td>0.09</td>
<td>15208</td>
</tr>
<tr>
<td>0.09</td>
<td>avaleht, inimesedtudengid</td>
</tr>
<tr>
<td>0.09</td>
<td>15205</td>
</tr>
<tr>
<td>0.08</td>
<td>avaleht, inimesedInstituudid</td>
</tr>
<tr>
<td>0.08</td>
<td>15213, inimesedtudengid</td>
</tr>
<tr>
<td>0.08</td>
<td>avaleht, 15213, inimesedtudengid</td>
</tr>
<tr>
<td>0.08</td>
<td>avaleht, 15208</td>
</tr>
<tr>
<td>0.07</td>
<td>15208, inimesedInstituudid</td>
</tr>
<tr>
<td>0.07</td>
<td>avaleht, 15208, inimesedInstituudid</td>
</tr>
</tbody>
</table>

Figure 4: Frequent closed itemsets having more than relative support=0.06. Support value is given with rounding error and closed itemsets were found using absolute support.
4 Frequent sequential patterns

Simply by listing all frequent item sets we will not see much of how users actually use the website. This is because of two reasons: firstly, frequent item sets do not capture the order of pages visited, and secondly, information about multiple page visits is lost when constructing item sets.

This problem can be solved by mining frequent sequential item sets. Sequential pattern mining is about mining frequent event sequences from databases, which means that we are searching for sequences of clicks that more often than not are followed by each other. Frequent sequences can be used to see how users actually tend to navigate and reach some desired resource.

![Graph showing the number of frequent sequential patterns depending on relative support.](image)

Figure 5: Number of frequent sequential patterns depending on relative support.

To mine frequent patterns from our dataset, we used an *a priori*-based frequent item sequence miner that uses a trie to store the candidates [Bod05]. Comparing to simple frequent itemsets, sequential patterns reveal more interesting results (Figure 6). The first thing to notice is a continuous reloading of the root page repeating over quite big time intervals. Our guess is that this is caused by the people who have set the faculty web site as their browser’s home page, which causes the faculty’s page to be loaded each time a new browser window is opened. We have no better explanation for this phenomenon as the main page is currently pretty static and is seldom updated.

The most important results from a usability point of view are the patterns where users visit page *inimesedInstituudid*. Users spend relatively little time on this page, for example in the sequence

\[
\text{avaleht} \rightarrow \text{inimesedInstituudid} \rightarrow 15208
\]
the median of time spent is only 3 seconds, but the next page the user requested is viewed for 30 seconds. The probable reason for this is that the page inimesedInstituudid is only a navigational page - it just links to other more important pages. Since users tend to spend only a few seconds choosing the right path, this page can probably be removed by merging the content of the pages it links to, or providing the links on a previous page.
<table>
<thead>
<tr>
<th>Support</th>
<th>frequent sequential pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63</td>
<td>avaleht</td>
</tr>
<tr>
<td>0.38</td>
<td>avaleht → avaleht</td>
</tr>
<tr>
<td>0.16</td>
<td>avaleht → avaleht → avaleht</td>
</tr>
<tr>
<td>0.16</td>
<td>ati</td>
</tr>
<tr>
<td>0.13</td>
<td>15213</td>
</tr>
<tr>
<td>0.1</td>
<td>15213 → 15213</td>
</tr>
<tr>
<td>0.1</td>
<td>inimesedtudengid</td>
</tr>
<tr>
<td>0.09</td>
<td>inimesedInstituudid</td>
</tr>
<tr>
<td>0.09</td>
<td>15208</td>
</tr>
<tr>
<td>0.09</td>
<td>avaleht → 15213</td>
</tr>
<tr>
<td>0.09</td>
<td>avaleht → inimesedtudengid</td>
</tr>
<tr>
<td>0.09</td>
<td>ati → ati</td>
</tr>
<tr>
<td>0.08</td>
<td>15205</td>
</tr>
<tr>
<td>0.08</td>
<td>inimesedtudengid → 15213</td>
</tr>
<tr>
<td>0.08</td>
<td>avaleht → inimesedInstituudid</td>
</tr>
<tr>
<td>0.08</td>
<td>avaleht → avaleht → avaleht → avaleht</td>
</tr>
<tr>
<td>0.08</td>
<td>inimesedtudengid → 15213</td>
</tr>
<tr>
<td>0.08</td>
<td>avaleht → 15213 → 15213</td>
</tr>
<tr>
<td>0.07</td>
<td>avaleht → 15208</td>
</tr>
<tr>
<td>0.07</td>
<td>inimesedtudengid → 15213 → 15213</td>
</tr>
<tr>
<td>0.07</td>
<td>inimesedInstituudid → 15208</td>
</tr>
<tr>
<td>0.07</td>
<td>avaleht → inimesedtudengid → 15213 → 15213</td>
</tr>
<tr>
<td>0.07</td>
<td>avaleht → 15205</td>
</tr>
<tr>
<td>0.07</td>
<td>avaleht → inimesedInstituudid → 15208</td>
</tr>
<tr>
<td>0.07</td>
<td>15213 → 15213 → 15213</td>
</tr>
<tr>
<td>0.06</td>
<td>oppetootunniplaanid</td>
</tr>
</tbody>
</table>

Figure 6: Frequent sequential patterns with time decorations. Time is given in seconds and is calculated by taking median value of time spent on that page over all sequences that contain the pattern.
4.1 Page access times

Because the log files contain time and date info for every request made by the user, we know the time of each mouse click the user made (to a precision of 1-2 seconds, taking into account the time it takes for a request to reach the server). From this we can find how much time users actually spend on a web page by calculating the time interval between two clicks. For every page, we can calculate the average time users spend on this page. However a more effective approach is to apply this method to previously found sequential patterns. Namely, for every frequent sequence found, we found all sessions that contain the sequence and calculated median time spent viewing each page in that sequence. We chose median time as we consider this more appropriate for experimental evaluation.

This method allowed us to determine the pages that are visited only because of navigation and not for content. For usability reasons, such pages should be removed to reduce user clicks.

5 Maximal reference sequences

Path traversal pattern mining has been studied extensively to find better alternatives to regular frequent sequence mining in the context of web mining. A solution has been proposed \cite{CPY98} that provides regular sequential patterns, but reduces the patterns only to forward reference sequences, that is patterns that only contain forward navigations. Authors of \cite{CPY98} say that forward references illustrate what people are looking for (e.g. destination pages) and eliminate back references from sessions.

When we applied this algorithm to our dataset, we found almost the same frequent patterns as with the regular sequential pattern mining technique. The difference was only the number of patterns found, which means that by eliminating back references and repeating pages, we find less patterns. When the number of patterns is bigger, this method helps to achieve better overview of frequent user traversal paths and target pages than the previously used method.

6 Changes in patterns through time

Pattern mining can also give insight into how patterns change over time. When mining school website data, one can hypothesize how users’ interest changes in time. We can presume, that since August is the pre-semester period and students might be looking for information about new professors, timetables or good seminars provided on the upcoming semester. Once the semester starts in September, it can be assumed that these kinds of aims become outdated and new behavioral patterns arise.
We split up the data set into two: August and September and compared sequential patterns found for both datasets with same relative support, $s=0.06$.

Results reveal that in August, users were seeking for general announcements. The page listing general news (\textit{yldteated}) had support of 0.073 in August (Figure 7), but in September (Figure 8) it was below 0.06.

In September, sequences containing patterns for the contact information pages of faculty employees (\textit{inimesedtudengid}, \textit{inimesedInstituudid} had 3\% higher support than in August. The page listing timetables (\textit{oppetootunniplaanid}) was more frequently accessed in August than in September.

Since the results somewhat confirm our assumptions, we conclude that this method can be used to compare user interests for different time periods.

\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|}
\hline
Support & frequent itemset \\
\hline
0.65 & avaleht \\
\hline
0.15 & 15213 \\
\hline
0.12 & inimesedtudengid \\
\hline
0.12 & avaleht, 15213 \\
\hline
0.11 & avaleht, inimesedtudengid \\
\hline
0.11 & 15213, inimesedtudengid \\
\hline
0.11 & oppetootunniplaanid \\
\hline
0.11 & avaleht, 15213, inimesedtudengid \\
\hline
0.1 & 15205 \\
\hline
0.07 & oppetoooppekavad \\
\hline
0.07 & inimesedInstituudid \\
\hline
0.07 & yldteated \\
\hline
\end{tabular}
\end{center}
\caption{Frequent itemsets for August.}
\end{table}
7 Restricting patterns by user specified pages

In our data set, the number of different pages visited is relatively big compared to the total number of sessions and average session length, and many interesting frequent patterns may not be visible without lowering the support threshold too much. A solution for overcoming this problem is to identify interesting pages and restrict frequent pattern mining algorithms to sessions containing these pages. The output should contain more relevant patterns with higher support. This method provides a simple way to analyze use cases more separately and thoroughly.

We conducted the experiment and focused only on sessions that contained the page inimesed/Instituudid. The results did not reveal anything interesting, but helped us find more patterns about the page than we had found before. For example, a frequent pattern with 40% support among filtered sessions was discovered that showed how in some cases users tend to reload the front page before arriving on inimesed/Instituudid.

8 Summary

We have shown a practical example of how traditional frequent pattern mining algorithms can be useful in web analytics context and understanding users’ need. Sequential patterns decorated with times spent on pages helped us identify hop-pages without even taking a look at the website.
More work in this field could be done in researching possible outliers detection and pattern significance measures to even more simplify interesting pattern detection.

References


