A User Profile Derivation Approach based on Log-File Analysis

Gerald Stermsek, Mark Strembeck, Gustaf Neumann
Institute of Information Systems, New Media Lab
Vienna University of Economics and BA, Austria
{firstname.lastname}@wu-wien.ac.at

Abstract - ***We present an approach to derive user profiles from observing user behavior. In particular, we use Web server log files and meta data describing page contents to extract user interests. In the derivation process, we apply descriptive statistics, like page access frequency, as well as more sophisticated means, such as network analysis methods. The approach requires that appropriate meta-data describing the content of web pages is available. However, we do not require a specific meta-data format or standard.***

Keywords: Log File Analysis, Personalization, User Profile

1 Introduction

The constantly growing information supply in Internet-based information systems poses high demands on concepts and technologies to support users to filter relevant information. *Information retrieval (IR)* and *information filtering (IF)* are two analytical information seeking stategies (see, e.g., [16, 20]) – in this paper we focus on information filtering.

Information filtering assumes a rather stable user interest (reflected through a user profile) but has to deal with highly dynamic information sources [20]. In IF systems, a user profile typically includes long-term user interests [3] and the acceptance of IF systems highly depends on the quality of user profiles (cf. [17]). In particular, a user profile describes a set of user interests which can be modeled via categories like sports, technology, or nutrition, and can be used for the purpose of information filtering. The definition of user profiles can either be explicit, implicit or a combination of both (cf. [11]). In the *explicit* approach the system interacts with the user and acquires feedback on information that the user has retrieved or filtered respectively. In turn, the user can, for example, indicate which filtering results are of most interest to him to improve future filtering results (so called relevance feedback).

The approach presented in this paper focuses the implicit acquisition and interpretation of user data. The *implicit* approach to user profile acquisition aims to derive user profiles by observing user behavior. In particular, we analyze the click-stream of users extracted from Web server access log files and the content-related meta data of the corresponding Web pages. By analyzing the user's log file records we try to deduce what the corresponding person is interested in and model his interests in a user profile. Our analysis includes descriptive statistics, such as page access frequency, as well as more sophisticated means, such as network analysis methods (cf. [12]).

Several approaches exist to analyze click paths on a Web site, concepts from graph theory (see, e.g., [7, 14]) or probabilistic approaches (see [21]) are most common. In this paper, we especially use graph theoretical concepts because they offer a number of metrics useful for eliciting user interests from click paths. In addition, if the user is prepared to explicitly provide his interests, we integrate this information with the implicitly generated user profile.

However, observing user behavior always raises privacy concerns. Therefore, we use P3P policies to keep users informed about the data we gather and about the purpose we use these data for. P3P [6] defines a standard way of encoding Web site privacy policies in a machine-readable XML format. In general, our P3P policies inform users that we collect several elements of Web server log files. And we only derive profiles for users who agree with these policies. However, this paper does not further elaborate on how we use P3P policies in our approach.

The remainder of this paper is structured as follows. Section 2 gives an overview of our approach before Section 3 describes the related sub-tasks in detail. In this paper we use the example of a soccer news Web site to explain the different steps of our approach. Furthermore, in Section 4 we describe a case study conducted with the access log files of the Institute of Information Systems at the Vienna University of Economics. Section 5 discusses related work and Section 6 concludes the paper.

2 Approach Overview

Typically, each user has diverse information interests related to his private or professional life which can be modeled via categories. Each of these categories depicts user interests on different abstraction layers whereas a tree is a natural way to represent these abstraction layers. Nodes in proximity of the root node indicate more general interests of an user while interests associated with leaf nodes represent specific topics. A tree also reveals a temporal view on user interests as general interest are rather stable over time and bottom-level interests are subject to more frequent changes (see also [10]).

The approach presented in this paper relies on a number of different types of information, some of which are mandatory and some are optional.

- *User-ID* (mandatory): identification of individual users.
- *Requested pages* (mandatory): reveals what information was accessed.
- *Content-related meta-data* (mandatory): semantic information about the information the user requested. An information provider may, for example, annotate the individual pages with meta-tags or use RDF-statements to describe the data.
- *User session information* (mandatory): this is particularly important to identify the number of significant pages for a specific user.
- *Structure information* (optional): information about the structure of the corresponding Web site.

In addition, to asses the content requested by the users we use several factors. These factors are:

- *Time* (t) indicates the time a user spents on a specific page.
- *Frequency* (f) indicates how often the user requested a specific page.
- *Centrality* (c) means that a page has short paths to all other nodes and thus is in a *central* position.
- *Prestige* (p) describes how often a page is referenced by other pages and thus has a prominent role in the graph of pages (see e.g. [12]).

These factors are weighted according the domain of the information system for which user profiling is conducted, and the information available (see Section 3.2.2). If, for example, no structural Information is available, the centrality and the prestige factors are weighted zero. The time factor is particularly important. Morita and Shinoda [19] identified a positive correlation between user interest and reading time and a low correlation between reading time and article length. Thus, we use the time spent on a page (TSP) as an indicator of interest but also have to consider user distraction. This effect is quite obvious as actions like answering the phone cause a longer TSP although the user is not actively looking at the page (see also [13]).

In practice, researchers tend to use heuristics gained from observing users, because it is impossible to identify user distraction on the server side. Typically a threshold is set for maximal (reasonable) page view time and extreme values are replaced by some standard view time calculated from the residual data (see [13]). Determining a threshold time, however, is not deterministic and often domain-driven. One possibility is to determine this threshold in absolute values, e.g. to replace values higher than 5 Minutes with the mean of the remaining values. Another option is to use relative values, for example, to declare values higher than three times the median value as extreme values and replace them with the mean of the residual values. Using relative values is a more flexible approach as the threshold automatically adapts to the respective domain.

A high level view of our approach is shown in Figure 1. The behaviour of Web users browsing from page to page is depicted as *steps* and results in a click path. Each step is represented by a node in the click path and refers to a Web page visited by the user. In particular, with every request the user generates an entry in the Web server's log file indicating his (potential) interest in that specific page and its content. Those log file entries are analyzed and form the basis to derive *user profile elements* which are then aggregated to a comprehensive user profile. A user may also provide his interests in an explicit manner by providing a (structured) list of categories. Such a list also constitutes elementary user profile elements and can be used to enhance an automatically derived user profile.

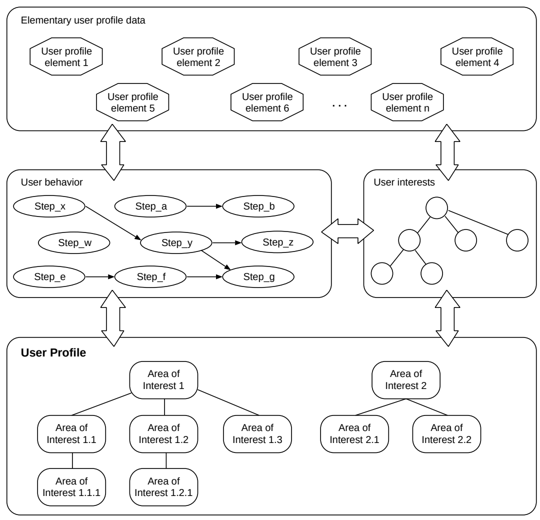

Figure 1. High level view of the approach

In general, user behaviour can be observed on client or on server machines. Tracking users on client machines may generate more and better data but requires active cooperation from users in contrast to tracking users on the server side by observing log files. Depending on the desired outcome we use log files from individual Web servers and/or Web proxies.

Figure 2 shows two possibile sources for collecting user profiling data from server log files. In Figure 2 a) data is gathered from individual Web sites, e.g. a sports news Web site, a tech news Web site, and a concert/event news Web site. In a typical usage scenario a user clicks his way through the pages he is interested in and Web servers generate corresponding log data which can be used to model his interests. This means, of course, that we would need access to all relevant Web server log files.

An alternative approach to get a comprehensive view on user interests is shown in Figure 2 b). In this scenario, all Web traffic is handled by a Web proxy and therefore every request, independent of its destination, can be

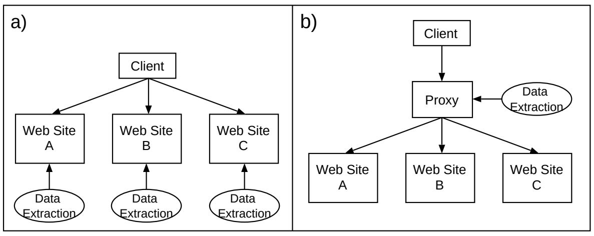

Figure 2. Possible request scenarios

recorded at a single point. Deriving a profile based on proxy log files gives a more complete view on user interests and thus a more complete user profile. However, using a proxy to record and analyze user behavior may also raise problems with privacy. As mentioned above, we thus use P3P policies to inform users about the information we gather.

3 User Profile Derivation

Using log file analysis for user profiling has a number of advantages (see, e.g., [22, 18]). For example, recording is done without bias and is technically easy. A disadvantage of log file analysis is that log files contain a lot of data which is possibly worthless for the purpose of user profiling and therefore log files have to be *preprocessed* (cleaned) prior to further processing measures (see Section 3.1). Another disadvantage is that if a user works with more than one instance of his browser or uses tabbed browsing, the corresponding click paths are shown in log files in strict chronical order and do not depict the parallel usage.

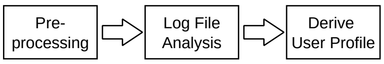

Figure 3. Major steps of the process

Moreover, because HTTP is a stateless protocol and does not provide session information, we use cookie logging on the server side which results in more valuable data for user profiling (cf. [15]).

The presented user profile derivation approach consists of three major steps, as depicted in Figure 3.

Figure 4 depicts the process in detail. After the identification of user sessions, statistical and graph analysis is applied on them. Then, the most significant pages are identified by integrating statistical (here *statistical analysis* means to analyze which pages the user visited most often, on which pages the user spent the most time, etc.) and graph analysis. Finally, those pages are integrated with content-related meta-data of the respective pages.

3.1 Preprocessing

Data preprocessing removes the data not important for analysis and prepares the log files for further processing. In particular, we extract the following tupel from the Web server log file:

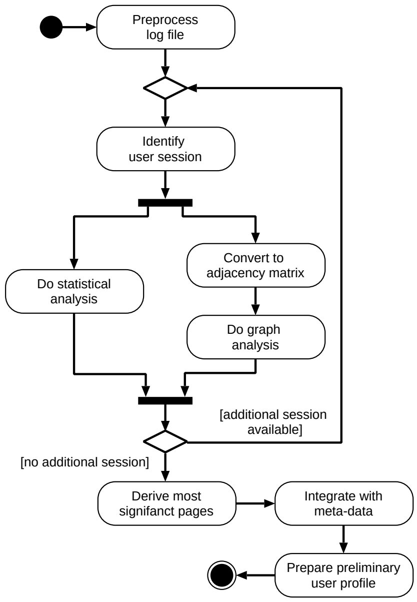

Figure 4. Log file processing

⟨page, page meta-data, ?referrer?, timestamp, cookie⟩

Here, referrer information is optional and reveals information about the corresponding Web site structure. In our approach, cookies especially include user IDs so that timestamp and cookie information can be used to identify user sessions. In particular, preprocessing includes the following main steps:

- *Extract page impressions:* For user profile derivation we only consider *page impressions*. A page hit is the request of an arbitrary file from a Web server. In contrast, page impression refers to the request of a complete Web page (e.g. including an HTML page and several images).

- *Extract page information:* We consider successful GET and POST request only. While failed requests may be relevant for the administrator of a Web site they are not relevant for user profiling and are eliminated prior analysis. Thus, we remove all log file entries with HTTP status codes other than 200 (successful request) or 304 (not modified). It is important to remove requests from Web robots as only human behavior should be modeled. Robots can be identified by their User-Agent HTTP request header, for example 'Googlebot' or 'Ahoy!'. Finally, the page information is compiled from the URL of the page, post-data (if available) and cookie information. Additionally, each relevant page is annotated with meta-data describing the content of the respective page.

- *Extract user and session information:* In our approach an individual user can be identified via cookie information. A *user session* can be defined as a group of activites by a single user whithin a specific time frame. Time frame identification is the most common method to identifiy sessions (see

[4, 5]). A threshold for maximal page view time (usually thirty minutes) is used to distinguish user sessions. Thus, we also use this standard threshold as an indicator of a new session.

The various levels of caching, be it on client side or by proxies also constitute a drawback of log file analysis. For example, hitting the 'back' button in a Web browser takes the user to the previously visited page but causes no request which is recorded in the log files. Those revisits, however, are an essential characteristic of navigation behavior as they often indicate which pages are observed as related (cf. [22]). To resolve the caching problem, Cooley suggests to exploit the knowledge of the site's organization (see [5]).

3.2 Log File Analysis

3.2.1 Overview

After log file preprocessing, we start the log file analysis. It is most common to use statistical techniques to extract knowledge about Web site users (see, e.g., [23]).

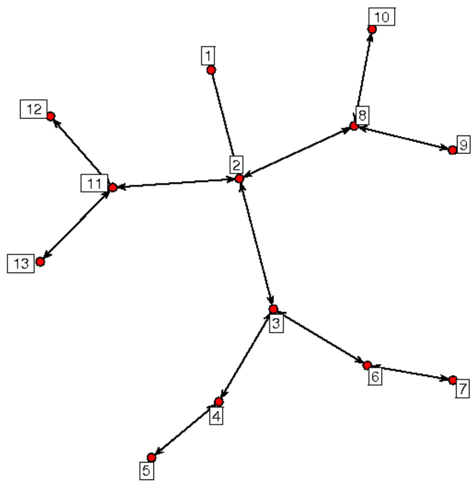

Figure 5. Structure of the example site

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]
[1,]	0	1	0	0	0	0	0	0	0	0	0	0	0
[2,]	0	0	1	0	0	0	0	1	0	0	1	0	0
[3,]	0	1	0	1	0	1	0	0	0	0	0	0	0
[4,]	0	0	1	0	1	0	0	0	0	0	0	0	0
[5,]	0	0	0	1	0	0	0	0	0	0	0	0	0
[6,]	0	0	1	0	0	0	1	0	0	0	0	0	0
[7,]	0	0	0	0	0	1	0	0	0	0	0	0	0
[8,]	0	1	0	0	0	0	0	0	1	1	0	0	0
[9,]	0	0	0	0	0	0	0	1	0	0	0	0	0
[10,]	0	0	0	0	0	0	0	1	0	0	0	0	0
[11,]	0	1	0	0	0	0	0	0	0	0	0	1	1
[12,]	0	0	0	0	0	0	0	0	0	0	1	0	0
[13,]	0	0	0	0	0	0	0	0	0	0	1	0	0

Figure 6. Matrix view of the example

For *graph analysis*, the log file is first converted to an adjacency matrix. An adjacency matrix is a matrix representation of the individual user requests and represents the usage pattern of a Web site for a certain user. This matrix shows which pages were requested by the user and how the user got to the different pages (we use the *request* and *referrer* information of the log file to construct the matrix and to draw the corresponding graph). The adjacency matrix is the starting point for graph analysis which identifies, for example, pages with high centrality. A graph representation of an example web site for soccer news is shown in Figure 5 and the respective matrix representation in Figure 6 (an excerpt of the corresponding log file is shown in Figure 7). Each matrix cell represents the number of links between two Web pages, whereas a zero indicates that no link exists between the pages.

The integration of statistical and graph theoretical figures is an important part of our user profile derivation approach.

3.2.2 Identify Significant Pages

For each user session we determine how many Web pages have been visited in this particular session. A typical Web site may consist of many individual pages which are of different interest to a user and therefore of different importance for his user profile. In particular, we just consider the s most significant pages out of a given population of n pages for each user session. However, it is also possible to consider all pages of the user session for user profiling and apply an adequate weighting scheme, which weights more significant Web pages higher than less significant ones in the resulting user profile.

Our assumption is that on a small special interest Web site each (or most) Web page(s) is/are of interest to the user and therefore significant for our analysis. But if the number of pages on a Web site increases not every page is equally important. We use the following formula to determine the number of significant pages: $s = 1 + 2 * log(n)$.

This formula is the result of an evolutionary process where we tested different formulas in an explorative data analysis to determine which formula delivers reasonable results. Using the logarithm we keep the number of significant pages s near the number of pages n for low numbers of n and keep s low for higher numbers of n. Our example page (see Figure 5) consists of thirteen nodes ($n = 13$), thus the number of significant nodes for this Web site is $s = 1 + 2 * log(13) = 6.13$. Which means that we consider six nodes out of thirteen to derive the user profile.

After we know how many pages are important for user profiling we have to determine which pages (nodes) are most significant. We begin our analysis with statistical methods and calculate the frequency of the individual pages and the time spent on each page. For example, we have to consider that a user can visit a specific Web page very often but spent just a short time on it because this page lies on the path to another page the user is interested in. For this reason we weight the Web page hit frequency lower in our final calculation (see below).

The accumulated time spent on each page is represented as a vector in Figure 8. In this example, we see a high value for the page `/soccer/leagues/` which could indicate user distraction. As mentioned in Section 2, we determine a threshold to eliminate extreme values. In particular, we use three times the median of the vector to formally identify such pages. This value

```
... "GET / HTTP/1.1" 200 44 "-" "Mozilla/5.0..."
... "GET /soccer/ HTTP/1.1" 200 901 "http://example.com/" "Mozilla/5.0..."
... "GET /soccer/cups/ HTTP/1.1" 200 831 "http://example.com/soccer/" "Mozilla/5.0..."
... "GET /soccer/cups/europeancup/ HTTP/1.1" 200 755 "http://example.com/soccer/cups/" "Mozilla/5.0..."
... "GET /soccer/cups/ HTTP/1.1" 200 831 "http://example.com/soccer/cups/europeancup/" "Mozilla/5.0..."
... "GET /soccer/cups/worldcup/ HTTP/1.1" 200 733 "http://example.com/soccer/cups/" "Mozilla/5.0..."
... "GET /soccer/cups/worldcup/wc2006/ HTTP/1.1" 200 642 "http://example.com/soccer/cups/worldcup/" "Mozilla/5.0..."
... "GET /soccer/cups/worldcup/ HTTP/1.1" 200 733 "http://example.com/soccer/cups/worldcup/wc2006/" "Mozilla/5.0..."
... "GET /soccer/cups/ HTTP/1.1" 200 831 "http://example.com/soccer/cups/worldcup/" "Mozilla/5.0..."
... "GET /soccer/ HTTP/1.1" 200 901 "http://example.com/soccer/cups/" "Mozilla/5.0..."
... "GET /soccer/clubs/ HTTP/1.1" 200 807 "http://example.com/soccer/" "Mozilla/5.0..."
... "GET /soccer/clubs/bayermunich/ HTTP/1.1" 200 601 "http://example.com/soccer/clubs/" "Mozilla/5.0..."
```

Figure 7. Example log file in Common Logfile Format

weighting	0.7		0.2		0.05		0.05			
	time	r	freq	r	centr	r	prest	r	total	total rank
/	0.005	13	0.042	7	0.02	13	0	13	0.012	13
/soccer/	0.057	8	0.167	1	0.208	1	0.174	1	0.092	5
/soccer/cups/	0.017	9	0.125	2	0.157	2	0.13	2	0.051	9
/soccer/cups/europeancup/	0.01	10	0.083	5	0.077	5	0.087	5	0.032	11
/soccer/cups/europeancup/ec08/	0.112	5	0.042	7	0.029	11	0.043	7	0.091	7
/soccer/cups/worldcup/	0.009	11	0.083	5	0.077	5	0.087	5	0.031	12
/soccer/cups/worldcup/wc2006/	0.16	2	0.042	7	0.029	11	0.043	7	0.124	2
/soccer/clubs/	0.005	12	0.125	2	0.136	3	0.13	2	0.042	10
/soccer/clubs/bayermunich/	0.119	4	0.042	7	0.033	7	0.043	7	0.096	4
/soccer/clubs/liverpool/	0.18	1	0.042	7	0.033	7	0.043	7	0.138	1
/soccer/leagues/	0.077	7	0.125	2	0.136	3	0.13	2	0.092	6
/soccer/leagues/bundesliga/	0.1	6	0.042	7	0.033	7	0.043	7	0.082	8
/soccer/leagues/premierleague/	0.148	3	0.042	7	0.033	7	0.043	7	0.115	3
sum	1		1		1		1			

Table 1. Summary of results for the soccer example

proved sensible in our tests but further investigation is required to verify that assumption for different domains. We assume the median is a better indicator for the typical page view time as the mean would be biased by high values. Due to experiences with our data, we further assume that three times the median is a high enough value to indicate distraction. In our example, the median is 111, thus values higher than 333 seconds represent pages with user distraction, which is the case with the `/soccer/leagues/` page. To normalize outliers we substitute them with the mean of the residual values. In our case, we replace 905 with the mean of the remaining 12 values, which is 85 (see bold line in Figure 8). Figure 8 also shows the normalized time vector. Normalization is required to make vectors comparable which represent different characteristics.

	seconds	revised	normalized
/	5	not changed	0.005
/soccer/	63	"	0.057
/soccer/cups/	19	"	0.017
/soccer/cups/europeancup/	11	"	0.010
/soccer/cups/europeancup/ec08/	124	"	0.112
/soccer/cups/worldcup/	10	"	0.009
/soccer/cups/worldcup/wc2006/	177	"	0.160
/soccer/clubs/	6	"	0.005
/soccer/clubs/bayernunich/	132	"	0.119
/soccer/clubs/liverpool/	199	"	0.180
/soccer/leagues/	**905**	**replace value: 85**	**0.077**
/soccer/leagues/bundesliga/	111	not changed	0.100
/soccer/leagues/premierleague/	163	"	0.148

Figure 8. Time spent on Web page vector

After statistical analysis, we analyze the structure of a Web site by using network analysis metrics. In a first step, we convert the log file to an adjacency matrix of dimension $n * n$ where n represents the number of Web pages (nodes) the Web site consists of. Based on this matrix we calculate distinctive network metrics such as centrality or prestige which give us an indication of important or prominent nodes of that Web site.

Table 1 summarizes the statistical and the graph analyses of our soccer example for a single user session. The individual Web pages of the example Web site are aligned line-by-line. The columns *time, freq, centr* and *prest* list the results of the individual valuations of Web pages with their respective rank. The first line of the table indicates the weighting scheme which leads to the total valuations shown in the column titled *total*. We calculate the total weighting through the following formula: $total = \alpha * t + \beta * f + \gamma * c + \delta * p$. The greek symbols are weighting factors while t, f, c and p represent the vectorized valuation of Web pages by time, frequency, centrality (which is itself an aggregated value) and prestige.

In our experience, the time factor is the most meaningful – moreover a positive correlation of time spent on a Web page and user interest has already been idenfied in [19]. The frequency factor is less important because, for example, a user may request a specific page just to get to another page that he is interested in. Centrality and Prestige introduce the network structure in the weighting scheme as more central nodes may have more significance. As weighting factors we suggest $t, f, c, p = 0.7, 0.2, 0.05, 0.05$ (see also Table 1).

This scheme is suggested on the basis of case studies conducted so far and constitutes a preliminary scheme. We assume that the weighting factors are domain dependent which is subject to further test series. Thus, the proposed weighting factors assume the frequency factor indicates *that* a user is interested in a specific page and the time factor denotes *how* interested the user is. After summing up the four columns in Table 1, we get a result vector which represents an aggregated valuation of all Web pages in the respective user session (*total* column in Table 1). From this result vector we take the first s nodes to form our user profile. Above we already determined $s = 6$, and those six significant pages are printed bold in Table 1.

3.3 Derive User Profile

User profile deriviation is the final step in our approach. In particular, we need semantic information (content-related meta data) concerning the most significant pages. Such semantic information can be gained in several ways:

- *Attributes:* A Web page can offer some of its internal semantic via meta-tags provided in the HTTP header, e.g. `<meta name="keywords" content="soccer, premiership, liverpool, player, peter crouch">`. Metadata can also be provided explicitly using RDF/XML statements [2].

- *Extraction:* If no explicit meta data is available text extraction and analysis may indicate relevant meta-data categories (cf. [25]). For example, analyzing the Web page $/soccer/clubs/liverpool/peter\_crouch.html$ may result to the following meta-data categories: `peter crouch, liverpool, soccer`.

- *File path:* When using certain Web development frameworks or for Web directories like Yahoo for example, it may be possible to use primitive heuristic to elicit some kind of structure from the Web page's file path, e.g. $/soccer/clubs/liverpool/peter\_crouch.html$

For the sake of simplicity, we assume that the relevant meta-data is available and can be used for user profile definition but do not determine a specific technique to define those meta-data.

3.3.1 Prepare User Session Information

We use the matrix representation of the Web site to form the user profile (see Figures 5 and 6). For this purpose we gather all page attributes associated with the individual Web pages (see Figure 9 a). This representation typically includes a lot of redundant information as more generic attributes are likely to occur quite often. Hence, we conduct a cleaning process to eliminate redundant attributes. The remaining attributes and their mutal relationships then give us a comprehensive view of the respective user session.

In a first step, we count the occurances of all attributes for the user session. In our example, this results in the following count: *soccer: 12, cups: 5, clubs: 3, leagues: 3, worldcup: 2, europeancup: 2, all remaining attributes: 1*. Afterwards, we remove all page attributes occuring more often than once starting with the most frequent attribute – unless the attribute is the last one remaining for the respective node. In our example (see Figure 9 a), this means that in a first step we remove the *soccer* attribute from all nodes except the root node as *soccer* is the only attribute of this node. In the next step, we remove the *cups* attribute from the respective nodes – unless *cups* is the only remaining attribute of this respective node. We continue this process until no more attributes can be removed. Attributes that are not removed in this process are printed in bold font in Figure 9 a).

The example above has several redundant attributes in its leaf nodes which can be removed. If an attribute occurs more than once in a sub-tree, however, the corresponding redundant node can be removed. If a Web page includes only a single attribute no cleaning is required. Note that the suggested user profile generation approach only works if proper Web page attributes are available. If, for example, all pages include exactly the same attribute(s) the approach fails.

3.3.2 Generate User Profile

As mentioned above, only the most significant Web pages are used for user profile generation (see bold entries in Table 1). As shown in Table 1 the highest rated Web pages are `2,7,9,10,11` and `13`. Those pages are drawn with thicker borders in Figure 9 a). To generate the respective user profile we pick out those nodes from the user session representation. In addition, we pick the parent nodes of the respective nodes to semantically revalue the derived user profile. The resulting user profile is shown in Figure 9 b).

While generating the user profile we assume that the log file entries (at least partially) reveal the site's organization to allow for drawing a preliminary interest tree. Otherwise the user profile results in an (possibly unstructured) list of user interests.

4 Experiences

To test our approach we analyzed the access log files of the Institute of Information Systems at the Vienna University of Economics. In particular, we evaluated the log files from Monday to Friday of calendar week 46/2006. We used the *R software environment* [1] for all statistical analyses.

The analysis started with 96059 log file entries which were then cleaned from spider requests and requests of

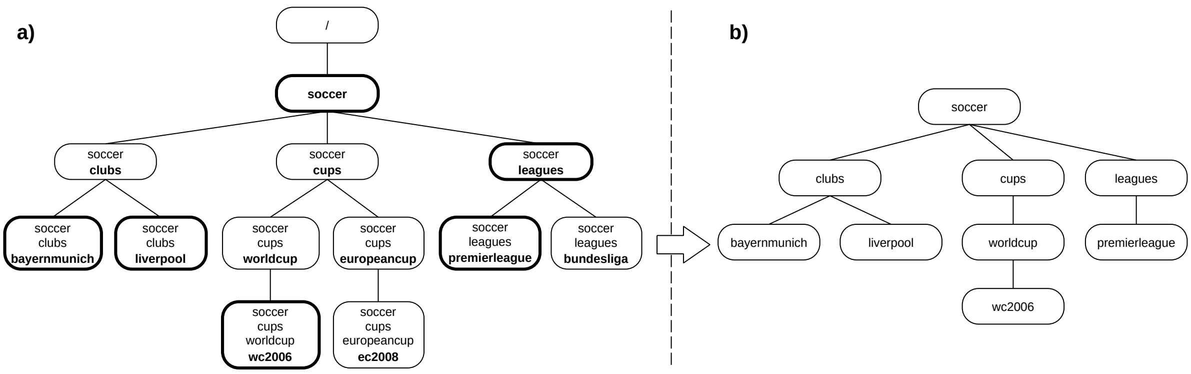

Figure 9. User profile generation

images loaded along with Web pages (see Section 3.1). During this procedure we identified 75 different spiders harvesting our institute Web site. We also removed HEAD requests, as HEAD requests are typically used to identify the content length of a file prior downloading. Technically they may also be used for caching purposes which is not easy to identify. However, due to the low number of HEAD requests (0.3 percent), we decided to remove them completely. We also identified a lot of requests with the HTTP status code 302 (moved temporarily) – this, however, results from the behavior of the site's CMS system which redirects request of `/main` to `/main/`, for example, which results in two log file entries, a request with status code 302 and a request with status code 200.

After preprocessing the log file still included 6895 entries (7.2 percent of 96059). This site uses cookies to identify users. About 77 per cent of these 6895 entries contained cookies. From these cookies we identified 1960 distinguishable users. We only used log file entries with cookie information for data analysis as they can be assigned to individual users unambiguously. Even a combination of IP address and user-agent information would be problematic, e.g. because of DHCP assigned IP addresses. The identified 1960 users caused 528 user sessions which consisted of 3 clicks or more. Hofgesang [13] suggested to remove sessions with less than 3 clicks, as most of them are accidental visits. The analysis of the 528 user sessions showed that the mean session length was 6.35 clicks, the standard deviation 6.08 and the maximum session length 43 clicks.

The valuation of the log file entries per user showed that in an average user session the user visited 4.60 Web pages – standard deviation 2.82, maximum value 23. From the 2430 pages the remaining users visited, we only consider the pages with high ratings – according to our algorithm, i.e. we considered 80.6 per cent (1960 pages) for user profile generation.

5 Related Work

Web usage mining (WUM) is the application of data mining techniques to large Web data repositories, some algorithms commonly used include association rule mining, sequential pattern generation, and clustering (see e.g. [5]). WUM produces aggregated results to better understand Web usage and improve the service provided to the customer (cf. [8]). In contrast, our approach concentrates on data mining at the level of the individual user resulting in non-aggregated data.

Early approaches only considered the popularity of Web pages in their analysis. In [19] Morita and Shinoda presented one of the first evaluations of the time aspect. Hofgesang [13] analyzed the factors incluencing the time spent on Web pages and combined the time and the frequency aspect to group users via clustering.

User profiling is frequently seen as a text classification problem. Thereby a supervised learning process is used to generate user profiles [24]. Supervised learning is a typ of machine learning and is characterized by the underlying learning model and its parameters. Depending on the formalisms provided by this learning model user profiles have complex internal structures and often serve as a blackbox to the users [10].

Ontology-based user profiling [9] uses ontologies to represent user interests via concept hierarchies. Compared to other concepts of user profile representation, e.g. vector space, using ontologies means a semantic revaluation (cf. [9]). General-purpose ontologies with a high number of concepts are often not appropriate for profiling a single user profile (cf. [10]). Ontologies often represent the shared knowledge of either a particular community or a group of users and therefore they may fail to capture an individual user's specific understanding of a domain [10].

6 Conclusion and Future Work

We presented a user profile derivation approach. In particular, we use Web server log files as data source for user profiling. The general approach, however, is applicable to arbitrary log file formats.

In particular, we use a combination of statistical analysis and graph analysis for user profiling. This allows for a broader perception of user behavior and has the potential to improve user profiling. In our future work, we will investigate possibilities to integrate automatically derived user profiles with explicitly provided user interests. Moreover, we will build a software tool that supports our approach.

References

[1] The R Project for Statistical Computing. http://www.r-project.org, 2006.

[2] Dave Beckett. RDF/XML Syntax Specification (Revised), W3C Recommendation, February 2004.

[3] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[4] Lara D. Catledge and James E. Pitkow. Characterizing browsing strategies in the World-Wide Web. *Computer Networks and ISDN Systems*, 27(6):1065–1073, 1995.

[5] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[6] Lorrie Cranor, Marc Langheinrich, and Massimo Marchiori. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation, April 2002.

[7] Canter D., Rivers R., and Storrs G. Characterizing user navigation through complex data structures. *Behavior and Information Technology*, 24(2):92–102, 1985.

[8] Yongjian Fu, Kanwalpreet Sandhu, and Ming-Yi Shih. A Generalization-Based Approach to Clustering of Web Usage Sessions. In *WEBKDD '99: Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, pages 21–38, London, UK, 2000. Springer-Verlag.

[9] Susan Gauch, Jeason Chaffee, and Alaxander Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent System*, 1(3-4):219–234, 2003.

[10] Daniela Godoy and Analia Amandi. User profiling for web page filtering. *IEEE Internet Computing*, 9(4):56–64, 2005.

[11] Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259, 2001.

[12] Robert A. Hanneman and Mark Riddle. *Introduction to social network methods*. University of California, Riverside, 2005. published in digital form at http://faculty.ucr.edu/ hanneman/.

[13] Peter Hofgesang. Relevance of time spent on web pages. In *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, 2006.

[14] Mark A. Horney. Case studies of navigational patterns in constructive hypertext. *Computers Education*, 20(3):257–270, 1993.

[15] D. Kristol and L. Montulli. HTTP State Management Mechanism. RFC 2965 (Proposed Standard), October 2000.

[16] Gary Marchionini. *Information seeking in electronic environments*. Cambridge University Press, New York, NY, USA, 1995.

[17] Alessandro Micarelli and Filippo Sciarrone. Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2-3):159–200, 2004.

[18] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, 2000.

[19] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 272–281, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[20] Douglas W. Oard. The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, 7(3):141–178, 1997.

[21] Ramesh R. Sarukkai. Link prediction and path analysis using markov chains. *Computer Networks*, 33(1-6):377–386, 2000.

[22] Myra Spiliopoulou. Web usage mining for web site evaluation. *Commun. ACM*, 43(8):127–134, 2000.

[23] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[24] Geoffrey I. Webb, Michael J. Pazzani, and Daniel Billsus. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):19–29, 2001.

[25] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.