USER PROFILING: WEB USAGE MINING

Miha Grčar
Department of Knowledge Technologies
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 31 657881; fax: +386 1 4251038
e-mail: miha.grcar@ijs.si

ABSTRACT

Web usage mining differs from collaborative filtering in the fact that we are not interested in explicitly discovering user profiles but rather usage profiles. When preprocessing a log file we do not concentrate on efficient identification of unique users but rather try to identify separate user sessions. These sessions are then used to form the so called transactions (see [3]). In the following stage, Web usage mining techniques are applied to identify frequent item-sets, sequential patterns, clusters of related pages and association rules (see Sections 3 and 4). Web usage mining can be used to support dynamic structural changes of a Web site in order to suit the active user, and to make recommendations to the active user that help him/her in further navigation through the site he/she is currently visiting. Furthermore, recommendations can be made to the site administrators and designers, regarding structural changes to the site in order to enable more efficient browsing. In the case of implementing Web usage mining system in the form of a proxy server, predictions about which pages are likely to be visited in near future can be made, based on the active users' behavior. Such pages can be pre-fetched to reduce access times.

1 INTRODUCTION

Web usage mining shows some similarities to collaborative filtering (collaborative filtering is discussed, for example, in [1]). If we consider pages to be items and we are able to efficiently identify users, we can perform collaborative filtering in order to provide the active user with recommendations about which pages he/she should also visit. Furthermore, we can point out links to pages that the active user will probably navigate to next. This approach, however, has several drawbacks. Each time the user accesses the site he/she may have different browsing goals. The user might prefer recommendations that focus on his current interest. Furthermore, the information about the sequential order of accesses is discarded in collaborative filtering. It is shown in [6] that this piece of information is significant for predictive tasks such as pre-fetching while it is less desirable for the recommendation tasks of collaborative filtering. Another problem arises when an efficient tracking mechanism based on user authentication and/or cookies is not available. In this case it is probably better to perform a variant of Web usage mining.

Since the user may have different browsing goals each time he/she accesses the site and since sessions are easier to identify in log files than users, sessions can be used as instances (instead of users). Each session is thus represented in the form of a feature vector as follows:

$s = (w_1, w_2, ..., w_n)$

where weight w_k is determined by the degree of the user's interest in the k-th page during session s, as described in Section 3.

We are now dealing with feature vectors, features being items (pages), just as in collaborative filtering. However, in this case vectors represent sessions and not users, which distinguishes Web usage mining from collaborative filtering.

2 WEB LOG DATA

Before going into algorithmic details of Web usage mining, let us briefly explain the Web log data preparation process.

Web logs are maintained by Web servers and contain information about users accessing the site. Logs are mostly stored simply as text files, each line corresponding to one access (i.e. one request). The most widely used log file formats are, implied by [15], the Common Log File format (CLF) [16] and the Extended Log File format (ExLF) [17]. The latter is customizable, which does not apply to CLF. The Web log contains the following information: (i) the user's IP address, (ii) the user's authentication name, (iii) the date-time stamp of the access, (iv) the HTTP request, (v) the response status, (vi) the size of the requested resource, and optionally also (vii) the referrer URL (the page the user "came from") and (viii) the user's browser identification. Of course, the user's authentication name is not available if the site does not require authentication. In the worst case, the only user-identification information included in a log file is his/her IP address. This introduces a problem since different users can share the same IP address and, what is more, one user can be assigned different IPs even in the same session.

2.1 Web Log Data Preparation

The data from Web logs, in its raw form, is not suitable for the application of usage mining algorithms. The data needs to be cleaned and preprocessed. The overall data preparation process is briefly described in the following sections.

2.1.1 Data Cleaning

Not every access to the content should be taken into consideration. We need to remove accesses to irrelevant items (such as button images), accesses by Web crawlers (i.e. non-human accesses), and failed requests.

2.1.2 Efficient User Identification

The user's IP address is but poor user-identification information [e.g. 4, 5]. Many users can be assigned the same IP address and on the other hand one user can have several different IP addresses even in the same session. The first inconvenience is usually the side-effect of intermediary proxy devices and local network gateways (also, many users can have access to the same computer). The second problem occurs when the ISP is performing load balancing over several proxies. All this prevents us from easily identifying and tracking the user. By using the information contained in the "referrer" and "browser" fields we can distinguish between some users that have the same IP, however, a complete distinction is not possible. Cookies can be used for better user identification. Users can block or delete cookies but it is estimated that well over 90% of users have cookies enabled [18]. Another means of good user identification is assigning users usernames and passwords. However, requiring users to authenticate is inappropriate for Web browsing in general.

2.1.3 Session Identification and Path Completion

Session identification is carried out using the assumption that if a certain predefined period of time between two accesses is exceeded, a new session starts at that point. Sessions can have some missing parts. This is due to the browser's own caching mechanism and also because of the intermediate proxy-caches. The missing parts can be inferred from the site's structure [3].

2.1.4 Transaction Identification

Some authors propose dividing or joining the sessions into meaningful clusters, i.e. transactions. Pages visited within a session can be categorized as auxiliary or content pages. Auxiliary pages are used for navigation, i.e. the user is not interested in the content (at the time) but is merely trying to navigate from one page to another. Content pages, on the other hand, are pages that seem to provide some useful contents to the user. The transaction generation process usually tries to distinguish between auxiliary and content pages to produce the so called auxiliary-content transactions (consisting of auxiliary pages up to and including the first content page) and the so called content-only transactions (consisting of only content pages). Several approaches, such as transaction identification by reference length [3] and transaction identification by maximal forward reference [3, 13] are available for this purpose.

3 ALGORITHMIC DETAILS

In this section we describe the algorithm for Web usage mining, presented in [7]. In the data preprocessing phase we extract a set of transactions [3]. Each transaction can be represented as a feature vector, features being pages and feature values being weights denoting the degree of the user's interest in a certain page during the transaction:

$t = (w_1, w_2, ..., w_n)$

Weights w_k can be defined in different ways. One of the possibilities is to represent them by the amount of time the user spends on a page. Note that the Web server has no exact notion of the time spent on a page. The duration of the visit can only be estimated from the time difference between two consecutive accesses. This approach seems reasonable, since it tends to weight content pages higher. However, it was observed in [7] that one long access can completely obscure the importance of other relevant pages. If we are dealing with transactions that do not contain navigational pages since these were filtered out, it is probably better to use other approaches. In such case, weights can be defined by the number of times a page was visited during the transaction. We can also simply use binary values stating "the page was visited at least once" and "the page was not visited".

Once a vector representation of transactions is obtained, we need to define a distance measure $d(t_1, t_2)$ between two vectors for the purpose of clustering the transactions. Cosine similarity measure can be used; the distance is in this case computed as $d(t_1, t_2) = 1 - \cos\theta(t_1, t_2)$, where θ is the angle between the two vectors t_1 and t_2. Another possibility is to use the Euclidean distance, computed as $d(t_1, t_2) = \sqrt{\sum_{i=1,...,n}(w_i^{(t_1)} - w_i^{(t_2)})^2}$. We can also define the distance measure by counting the number of overlapping non-zero weights n_{ovrl} in both vectors; $d(t_1, t_2) = 1 - n_{ovrl}/n$. The latter measure is used when comparing the active user's current session to the cluster medians, as described later on in the text.

In the next step, an unsupervised clustering algorithm is employed to discover different usage profiles. There are several well-known approaches that can be employed for this tasks, such as the leader algorithm, k-means algorithm, fuzzy membership approaches, Bayesian clustering, etc. Once clusters are obtained, a median transaction can be computed for each cluster:

$\bar{t} = (\bar{w}_1, \bar{w}_2, ..., \bar{w}_n)$

The main characteristics of a cluster are evident from its median transaction. Pages with higher median weights contribute more to the nature of the cluster.

The active user's current session (referred to as an active session) is maintained in the form of a vector. Each time the user requests a new page, the vector is updated and compared to cluster medians in order to find the cluster in which the user's current browsing behavior can be categorized. Not all similarity measures are equally successful in this task. In [7] weights are computed by counting the number of overlapping non-zero weights (denoted by n_{ovrl}) in both vectors (the active session vector and cluster median $\bar{t}$) and applying the following distance formula:

$$d(s_a, \bar{t}) = 1 - \frac{n_{ovrl}}{n}$$

In the following step, medians and thus clusters that are very similar to the active session (this means that the distance is below some predetermined threshold) are discovered. Pages in these clusters that have high median weights and are not contained in the active session are then recommended to the user. An additional weighting can be done to reward pages that are farther away from the active session with respect to the site's structure. These recommendations tend to be more interesting, since they are providing shortcuts to other (distant) sections of the site. Other more sophisticated methods for providing interesting recommendations have also been discussed [11].

Some authors argue that clustering based on distance measures is not the most prospective approach [10]. They state that the similarity (distance) computation is not a trivial task since vector representations are usually not good behavioral indicators when it comes to Web transactions. They propose a slightly different approach involving association rules discovery. These approaches are discussed in the next section.

4 ASSOCIATION RULE DISCOVERY IN WEB USAGE MINING

Some authors find the association rules discovery approach to be more prospective than the approach discussed in Section 3.

After transactions are detected in the preprocessing phase, frequent item-sets are discovered using the A-priori algorithm [e.g. 12]. The support of item-set I is defined as the fraction of transactions that contain I and is denoted by $\sigma(I)$. Given two item-sets X and Y, the association rule can be expressed as $\langle X \Rightarrow Y, \sigma_r, \alpha_r \rangle$, where σ_r is the support of $X \cup Y$ and α_r is the confidence of the rule given by $\sigma_r/\sigma(X)$.

Frequent item-sets and their corresponding association rules are represented in the form of a hypergraph. A hypergraph is an extension of a graph where each hyperedge can connect more than two vertices. A hyperedge connects URLs within a frequent item-set. Each hyperedge is weighted by the averaged confidence of all the possible association rules formed on the basis of the frequent item-set that the hyperedge represents. The hyperedge weight can be perceived as a degree of similarity between URLs (vertices).

Since the hyperedge weight can be interpreted as a degree of similarity between vertices, the hypergraph can be partitioned into clusters using the hypergraph partitioning methods [e.g. 14].

Clusters formed in this way are examined to filter out vertices that are not highly connected to the rest of the vertices in the cluster. The measure for determining the degree of connectedness between vertex v and cluster c is defined as follows:

$$\text{conn}(v, c) = \frac{|\{\text{edge} : \text{edge} \subseteq c,\ v \in \text{edge}\}|}{|\{\text{edge} : \text{edge} \subseteq c\}|}$$

This equation measures the percentage of edges within the cluster that vertex v is associated to. A high degree of connectedness indicates that v is connected to many other vertices in the partition and is thus highly connected to the partition.

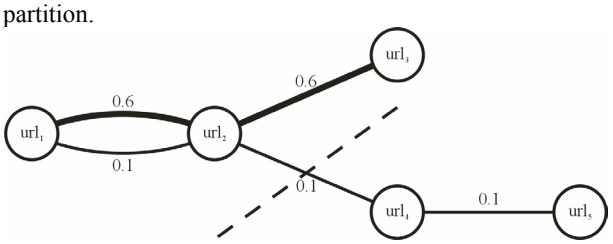

Figure 1: *A simple hypergraph. It consists of two hyperedges representing two frequent item-sets, namely {url_1, url_2, url_3} and {url_1, url_2, url_4, url_5}. Let us say, for example, that all possible association rules derived from the first item-set have the following confidence values (noted above the "implies" symbol): {url_1} $\overset{0.8}{\Rightarrow}$ {url_2,url_3}, {url_1,url_2} $\overset{0.4}{\Rightarrow}$ {url_3}, {url_1,url_3} $\overset{0.6}{\Rightarrow}$ {url_2}, {url_2} $\overset{0.4}{\Rightarrow}$ {url_1,url_3}, {url_2,url_3} $\overset{0.8}{\Rightarrow}$ {url_1} and {url_3} $\overset{0.6}{\Rightarrow}$ {url_1,url_2}. In this case the average confidence value – and thus the hyperedge weight – is 0.6. In this example, the other hyperedge has a weight of 0.1. If we now wish to partition this hypergraph into two clusters, we need to cut one or more hyperedges so that there are no interconnections between the two clusters. The cost of the partitioning is the sum of the weights of all the hyperedges that are cut in the process. We need to minimize this cost to make the partitioning reasonable. If we cut, for example, between vertices url_1 and url_2, the cost is 0.6 + 0.1 = 0.7. The lowest cost is achieved by cutting the hyperedge between url_2 and url_4, or between url_4 and url_5 (the cost is 0.1). The first cut gives us a more balanced partitioning, so it is best to cut the hyperedge between url_2 and url_4 (the dashed line). This gives us two clusters, namely {url_1, url_2, url_3} and {url_4, url_5}. In the first cluster, url_1 and url_2 are more strongly connected to the cluster than url_3 (see the definition of the connectedness function, conn(v, c)). Whether we filter url_3 out or not, depends on our choice of the threshold.*

To maintain the active session, a sliding window is used to capture the most recent user's behavior. The window size is determined by the average transaction size, estimated during the pre-processing phase. At each step, the partial

active session is matched with the usage clusters. Each cluster is represented in the form of a vector:

$c = (u_1^{(c)}, u_2^{(c)}, ..., u_n^{(c)})$

where u_k is the weight associated with the k-th URL (url_k) in the following way:

$$u_k^{(c)} = \begin{cases} conn(url_k, c), & url_k \in c \\ 0, & \text{otherwise} \end{cases}$$

The partial active session is represented as a binary vector $s = (s_1, ..., s_n)$, where $s_k = 1$ if the user accessed url_k in this session, and $s_k = 0$, otherwise. The next step is to compute the cluster-session matching score, match(s, c). In [10] the following equation is presented:

$$match(s, c) = \frac{\sum_k u_k^{(c)} \cdot s_k}{|s| \cdot \sqrt{\sum_k (u_k^{(c)})^2}}$$

After matching clusters are determined, the final step is to compute a recommendation score for URL u contained in a matching cluster, according to session s:

$$Rec(s, u) = \sqrt{conn(u, c) \cdot match(s, c)} \cdot ldf(s, u)$$

where an additional weighting is done to reward pages that are farther away from the active session with respect to the site's structure (incorporated with the so called link distance factor, ldf(s, u) [see 10]). URLs with high recommendation scores are recommended to the active user.

4.1 Incorporating Sequential Order of Accesses

Sequential order of the accesses in transactions is an important piece of information, mainly for the pre-fetching task. The association rules discovery approach to Web usage mining can be extended with the ability to detect frequent traversal patterns (termed large reference sequences) rather than frequent item-sets [13]. Other steps of this approach are modified accordingly, but are similar to the steps of the approach described in Section 4.

5. ACKNOWLEDGEMENTS

I would like to thank Marko Grobelnik and Dunja Mladenić for their mentorship and to Tanja Brajnik for every minute she invests in my English.

References

[1] J. S. Breese, D. Heckerman, C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. 1998.

[2] M. Eirinaki, M. Vazirgiannis. Web Mining for Web Personalization. *ACM Transactions on Internet Technology*. Vol. 3. No. 1. pp. 1–27. 2003.

[3] R. Cooley, B. Mobasher, J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. *Journal of Knowledge and Information Systems*. Vol. 1. No. 1. pp. 5–32. 1999.

[4] J. Pitkow. In Search for Reliable Usage Data on the WWW. *Proceedings of the Sixth International WWW Conference*. 1997.

[5] M. Rosenstein. What is Actually Taking Place in Web Sites: E-Commerce Lessons from Web Server Logs. *ACM Conference on Electronic Commerce*. 2000.

[6] B. Mobasher, H. Dai, T. Luo, M. Nakagawa. Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks. *Proceedings of the IEEE International Conference on Data Mining*. 2002.

[7] T. W. Yan, M. Jacobsen, H. Garcia-Molina, U. Dayal. From User Access Patterns to Dynamic Hypertext Linking. *Proceedings of the Fifth International World Wide Web Conference*. 1996.

[8] J. Srivastava, R. Cooley, M. Deshpande, P.-N. Tan. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*. Vol. 1–2. 2000.

[9] O. Nasraoui, H. Frigui, A. Joshi, R. Krishnapuram. Mining Web Access Log Using Relational Competitive Fuzzy Clustering. *Proceedings of the International Conference on Knowledge Capture*. pp. 202–208. 2001.

[10] B. Mobasher, R. Cooley, J. Srivastava. Creating Adaptive Web Sites Through Usage-Based Clustering of URLs. *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange*. 1999.

[11] R. Cooley, P.-N. Tan, J. Srivastava. Discovery of Interesting Usage Patterns from Web Data. *Proceedings of the International Workshop on Web Usage Analysis and User Profiling*. pp. 163–182. 1999.

[12] R. Agrawal, T. Imielinski, A. Swami. Mining Association Rules between Sets of Items in Large Databases. *Proceedings of the 1993 ACM SIGMOD Conference*. 1993.

[13] M.-S. Chen, J. S. Park, P. S. Yu. Data Mining for Path Traversal Patterns in a Web Environment. *Proceedings of the 16th International Conference on Distributed Computing Systems*. pp. 385. 1996.

[14] E.-H. Han, G. Karyps, V. Kumar, B. Mobasher. Clustering Based on Association Rule Hypergraphs. *Proceedings of Workshop on Research Issues in Data Mining and Knowledge Discovery*. 1997.

[15] Netcraft. Netcraft Web Server Survey. http://www.netcraft.com/survey/archive.html

[16] W3C. Logging Control in W3C httpd. http://www.w3.org/Daemon/User/Config/Logging.html

[17] W3C. Extended Log File Format. *W3C Working Draft WD-logfile-960323*. http://www.w3.org/TR/WD-logfile.html

[18] P. Baldi, P. Frasconi, P. Smyth. Modelling the Internet and the Web. pp. 171–209. ISBN: 0-470-84906-1. 2003.