Intelligent Techniques for Web Personalization and Recommender Systems
AAAI Technical Report WS-12-09

Capturing Browsing Interests of Users into Web Usage Profiles

Shaily Kabir, Sudhir P. Mudur and **Nematollaah Shiri**
Department of Computer Science and Software Engineering
Concordia University
Montreal, Quebec, H3G 1M8
{sh_kabir, mudur, shiri}@cse.concordia.ca

Abstract

We present a new weighted session similarity measure to capture the browsing interests of users in web usage profiles discovered from web log data. We base our similarity measure on the reasonable assumption that when users spend longer times on pages or revisit pages in the same session, then very likely, such pages are of greater interest to the user. The proposed similarity measure combines structural similarity with session-wise page significance. The latter, representing the degree of user interest, is computed using frequency and duration of a page access. Web usage profiles are generated using this similarity measure by applying a fuzzy clustering algorithm to web log data. For evaluating the effectiveness of the proposed measure, we adapt two model-based collaborative filtering algorithms for recommending pages. Experimental results show considerable improvement in overall performance of recommender systems as compared to use of other existing similarity measures.

1 Introduction

The World Wide Web is considered as the most dominant and enormous source of information most people interact with these days. Adapting a website through personalization is a common approach to improve interaction quality. Recommender systems, tools for Web personalization, attempt to provide personalized services to the users by recognizing their interests, captured as usage profiles. The performance of recommender systems largely depends on how efficiently the user preferences, interests, and goals are being realized [Castellano et al. 2010]. Among all recommender systems, collaborative filtering (CF) is the most popular one. Traditional CF systems (i.e., memory-based CF systems) produce recommendations for a target user by matching their current browsing patterns with preferences of a set of like-minded users. The entire logged data set is searched. The model-based CF approach has been evolved due to scalability and sparsity problems of the memory-based CF systems [Wang, Zhang, and Yin 2010]. It typically incorporates web usage mining (WUM) to develop an access behavior model based on past user browsing behavior, and employs this model to recommend pages to active users. An access behavior model can be generated in the form of a set of usage profiles by clustering web users based on their browsing interests. Therefore, successful clustering of users is the key to generating effective usage profiles. Clustering depends very much on the similarity measure defined between the items to be clustered. In this context, we propose a new *weighted similarity measure* which computes similarity among usage sessions by utilizing both session-wise page significance and their structural similarity. The page significance captures user interests and structural similarity captures closeness of the topics in the pages. Page significance takes into account both the time spent and the number of times a page is visited, while structural similarity is computed by utilizing their URL hierarchy structure. A set of usage profiles are generated using this weighted session similarity by applying the fuzzy clustering technique [Suryavanshi, Shiri, and Mudur 2005] to cleaned and sessionized web log data. For evaluating the effectiveness of our similarity measure, we have adapted two model-based CF algorithms, and conducted extensive experiments. Our results indicate that the proposed similarity measure improves the overall performance of recommender systems by providing higher quality recommendations compared to other similarity measures.

The rest of this paper is organized as follows. Section 2 reviews previous work on similarity measures and their use in creating web usage profiles through clustering. Section 3 presents web data preprocessing and page-significance measure. In Section 4, we introduce our weighted session similarity measure. Section 5 presents usage profile generation through fuzzy clustering. Section 6 describes our adaptation of model-based CF algorithms. Section 7 presents results from various experiments. Section 8 concludes the paper along with future work.

2 Background and Related Work

Establishing the right similarity metric to capture the browsing interests of the users is crucial for grouping users. Considerable research has been conducted to establish similarity among users based on their browsing data [Pierrakos et al. 2003]. One way is for the user to give a numeric rating to a page to show her/his interest [Deshpande and Karypis 2004]. Another way is to infer user interests by observing his/her access behavior from the web log data, such as time spent

on pages and/or page-visit frequencies, or access sequence [Xiao et al. 2001; Liu and Keselj 2007]. Two users are said to be similar and should be in the same cluster, if they possess similar browsing interests. Some research works give importance to the number of similar pages and their visiting order in computing similarity, while others pay attention to access frequency and/or duration. [Xiao et al. 2001] proposed a matrix-based clustering method for grouping users, where similarity between two users is computed by measuring the cosine of the angle between two vectors of common pages, considering access-ordering, access-frequency and viewing time separately. [Shahabi et al. 1997] created user profiles by capturing the links selected by users and took into consideration order of pages, viewing time, and cache references, using a JAVA remote agent. They computed similarity among users by taking the cosine similarity of their navigation paths, and grouped the users based on a path-mining algorithm. [Castellano et al. 2007] used access-time as showing user interest in a page, and generated a fuzzy set of access-time using two time thresholds. After measuring user similarity by fuzzy Jaccard coefficient, they used CARD+ algorithm to generate user profiles. Similarly, [Wang, Xu, and Wu 2008] developed a fuzzy multiset to characterize user interests by integrating page-click rate, viewing-time, and user preference, and applied max-min approach to compute fuzzy user similarity. They proposed a fuzzy multiset-based clustering algorithm (CAFM) to group pages and users as well. [Xue et al. 2005] generated clusters using K-means algorithm, while similarity between user ratings was measured by Pearson correlation-coefficient. They used clusters for smoothing unrated items of individual users and for selecting neighborhood to make predictions in a hybrid CF system. An important point to note is that all of these similarity computations are based only on common pages between the two usage sessions being compared.

It is well-known that inclusion of website structural information or prior domain knowledge with web usage data provides better user similarity [Bose et al. 2006]. Typically, pages in a website are organized according to a hierarchical relationship based on their content (topic). [Nasraoui et al. 1999] quantified this relationship among page URLs as a distance measure and incorporated it into a session similarity measure. For this, they considered each usage session as a binary vector of accessed pages having equal degree of user interest. Later, [Li and Lu 2007] incorporated page similarity computed using URL-similarity and viewing-time similarity in a sequence alignment method for assessing user similarity. In this paper, we propose a new weighted similarity measure by integrating user interest in pages with URL-similarity of pages.

3 Web Data Preprocessing

Fig. 1 shows the dataflow diagram of our personalization system with three major modules: (i) Web log preprocessing(WLP), (ii) Web Usage Mining(WUM), and (iii) Recommendation. The WLP module deals with session extraction, page-significance measure, and weighted session creation. The WUM module handles weighted session similarity computation and usage profile generation. The Recommendation module recommends a list of pages to an active user.

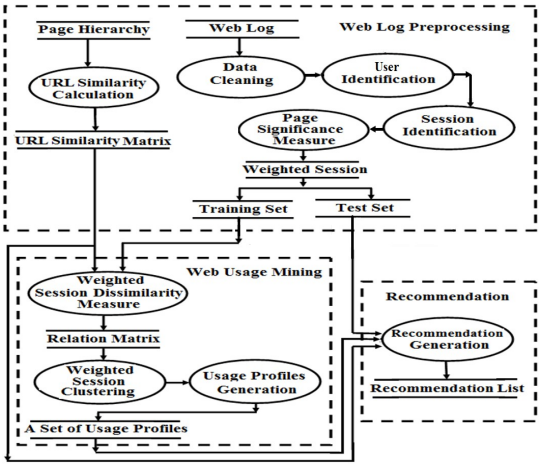


Figure 1: Dataflow diagram of personalization system.

From a given web log data, we first find basic usage information, which includes user IP address, requested date and time, requested URL, HTTP status code and file-size in bytes. After cleaning irrelevant entries such as requests from web robots and image entries, we use the IP address to group requests of individual users. We apply two time-oriented heuristics for session extraction [Spiliopoulou et al. 2003; Liu and Keselj 2007]. They are session-duration heuristic and page-stay time heuristic. We use 30 minutes as a threshold (θ) for session duration, and 10 minutes as a threshold (β) for page-stay time. A new session is considered when either θ or β is exceeded.

Let U=$\{\text{url}_1,\text{url}_2,...,\text{url}_M\}$ be the set of M pages in the website under study. Each usage session comprises a sequence of a subset of U together with access duration, frequency, and size. It is represented as $\text{us}_K=\{(\text{url}_1,\text{t}_1,\text{f}_1,\text{s}_1), (\text{url}_2,\text{t}_2,\text{f}_2,\text{s}_2), ..., (\text{url}_M,\text{t}_M,\text{f}_M,\text{s}_M)\}$, where url_j, t_j, f_j, and s_j are the visited page, it's total access-time in seconds, total frequency, and size (# of bytes), respectively.

3.1 A Measure for Degree of User Interest

While browsing, a user finds some pages more interesting than others. But the latter may be visited for other reasons like navigation, accidental visit, casual exploration, etc. This shows that pages accessed are of varying degrees of interests to a user. It is reasonable to assume that frequency and duration of accesses are two major indicators of a user interest in a page [Chan 1999; Liu and Keselj 2007]. Inspired by this, we propose a page-significance weighted measure for estimating the user interest, described below.

Page access duration does indicate degree of user interest, but it must also depend on the page content. A quick move to another page might be due to the page's size being small. Therefore, a user interest in a page of a session by means of "duration" can be estimated based on the time spent on a page with respect to it's size. This is further normalized by the maximum ratio in the session to recognize page importance compared to other pages. Equation (1) measures user

interest in a page url_j of a session us_K as regards "duration", where $0 \leq \text{Duration}_{url_j} \leq 1$.

$$Duration_{url_j} = \frac{\frac{\sum access\_time_j}{page\_size_j}}{max(\forall_{r \in us_K} \frac{\sum access\_time_r}{page\_size_r})} \quad (1)$$

Again, a user may go back to revisit a page in a single session arising from increased interest in the content. Hence, the user interest associated with a page in a session using "frequency" can be measured based on the number of visits normalized by the maximum number of visits in that session. Equation (2) measures user interest in url_j of a session us_K as regards "frequency", where $0 \leq \text{Frequency}_{url_j} \leq 1$.

$$Frequency_{url_j} = \frac{\sum visit_j}{max(\forall_{r \in us_K} \sum visit_r)} \quad (2)$$

Giving equal importance to the access duration and frequency, we define our page-significance measure as the harmonic mean of Duration_{url_j} and Frequency_{url_j}. We use harmonic mean since it tends to mitigate the impact of large outliers and aggravate the impact of small ones. Equation (3) shows the formula of "page significance" for url_j in us_K, where $0 \leq \text{Sig}_{url_j} \leq 1$.

$$Sig_{url_j} = \frac{2 \times Duration_{url_j} \times Frequency_{url_j}}{Duration_{url_j} + Frequency_{url_j}} \quad (3)$$

3.2 Weighted Usage Session Conversion

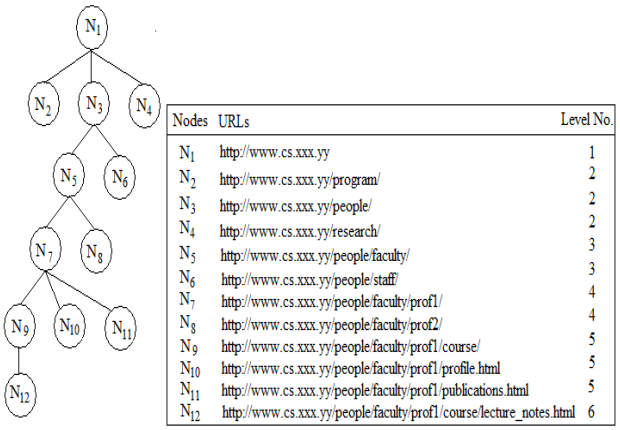
Let N be the number of extracted sessions. For generating weighted sessions, we measure session-wise significance of all pages using equations (1) to (3). Further, the most significant page is given rank 1 and the remaining pages are ranked accordingly. We term each session with a set of pages together with their significance and rank as "***Weighted Session***". It is presented as ws_K={(url_1,Sig_{url_1},rk_1), (url_2,Sig_{url_2},rk_2), ..., (url_M,Sig_{url_M},rk_M)}, where url_j, Sig_{url_j}, and rk_j denote the visited page, its significance, and rank, respectively.

4 Proposed Similarity Measure

In what follows, we first describe our proposed URL-based structural similarity measure for pages and then give the formulation for weighted session similarity computation which includes both URL-similarity and page significance.

4.1 URL-based Page Similarity Measure

The URL structure of websites is hierarchical. The intention is to assist users to narrow down into a topic. Each non-leaf node belongs to a page-URL corresponding to a directory of Web server. Each leaf node represents a page-URL that corresponds to a file. The root corresponds to the URL of home page of the website. We consider the root at level "one" of the hierarchy, L_1. Any non-leaf node at level L_k is directly linked to all of it's children at the immediate next lower level L_{k+1} via individual edges. This may be assumed to imply a "Consists-of" relationship between them. Fig. 2 shows a part of page hierarchy of a "CS Department" website.



Nodes	URLs	Level No.
N_1	http://www.cs.xxx.yy	1
N_2	http://www.cs.xxx.yy/program/	2
N_3	http://www.cs.xxx.yy/people/	2
N_4	http://www.cs.xxx.yy/research/	2
N_5	http://www.cs.xxx.yy/people/faculty/	3
N_6	http://www.cs.xxx.yy/people/staff/	3
N_7	http://www.cs.xxx.yy/people/faculty/prof1/	4
N_8	http://www.cs.xxx.yy/people/faculty/prof2/	4
N_9	http://www.cs.xxx.yy/people/faculty/prof1/course/	5
N_{10}	http://www.cs.xxx.yy/people/faculty/prof1/profile.html	5
N_{11}	http://www.cs.xxx.yy/people/faculty/prof1/publications.html	5
N_{12}	http://www.cs.xxx.yy/people/faculty/prof1/course/lecture_notes.html	6

Figure 2: A portion of page hierarchy of "CS Department" Website.

Generally, web pages sharing similar subjects are structurally related by URLs. Looking at the paths leading to pages from the root, it is possible to discover similarity among pages. This is useful in capturing subject similarity in user interests. In this context, [Nasraoui et al. 1999] defined a syntactic URL similarity measure for pages with the consideration that larger overlap in URLs must result in a higher similarity between pages. They used a URL-similarity of "one" for any node and its parent, and also for sibling nodes sharing the same parent. However, we believe that it is better to not assume a similarity of "one" for the URL-pairs (i.e., parent-child pair or siblings pair) located at different levels of a page hierarchy. In fact, more specialized information is likely to be derived from lower-level nodes compared to upper-level nodes. Moreover, as one dips more into a hierarchy, the topics in the URLs are conceptually related more. Therefore, we argue that any sibling/parent-child URL pairs positioned at deeper level(s) should possess greater similarity than those pairs at upper level(s).

Considering this, we define a URL-based similarity measure among pages based on their positions in a page hierarchy. The proposed URL similarity has three important features. Firstly, any URL pair with more overlap (longest common prefix) in the hierarchy possess higher similarity than any other URL pair with lesser overlap. Secondly, any pair of sibling URLs at L_n has greater similarity than any sibling pair at L_k when k<n. Lastly, any URL at L_n and it's parent URL is more similar than any URL at L_k and it's parent, when k<n. Our proposed URL-based page similarity measure for url_i and url_j is defined as follows:

$$URL_{sim}(i,j) = \frac{L(url_i \cap url_j)}{max(L(url_i), L(url_j))} \quad (4)$$

Here $\text{L}(\text{url}_i)$ is the level of node N_i, and $\text{L}(\text{url}_i \cap \text{url}_j)$ is the level of common ancestor node of url_i and url_j. Our URL structural similarity measure satisfies the following properties: (1) URL_{sim}(i,j)=URL_{sim}(j,i) and (2) $0 < \text{URL}_{sim}(i,j) \leq 1$.

As an example, let us consider the website structure (URL-based) shown in Fig. 2. Using equation (4), we

obtain $URL_{sim}(N_3,N_9)$ =0.40 and $URL_{sim}(N_7,N_{12})$=0.67. This formulation yields similarity values such that URLs with larger overlap do show higher similarity. In addition, $URL_{sim}(N_5,N_6)$=0.67 and $URL_{sim}(N_9,N_{10})$=0.80. This shows that sibling URLs at deeper level have higher similarity than any sibling pair at upper level. Also, $URL_{sim}(N_3,N_5)$=0.67 and $URL_{sim}(N_9,N_{12})$=0.83, i.e., parent-child URLs at deeper level are more similar than a parent-child pair at upper level.

4.2 Weighted Session Similarity Measure

The weighted session similarity **WSS** is defined as the maximum of two other measures: cosine similarity, $\mathbf{WSS}_1$ and structure-based cosine similarity, $\mathbf{WSS}_2$ to assess similarity among weighted sessions. The $\mathbf{WSS}_1$ determines cosine similarity between sessions, based on the significance of identical pages and completely ignores structural relation of pages. According to $\mathbf{WSS}_1$, a similarity of "one" is assigned for identical sessions with equal page-significance, but similarity score may vary with difference in significance. A similarity score of "zero" is assigned when the pages are different, independent of their positions in the page hierarchy. Equation (5) shows $\mathbf{WSS}_1$ for two weighted sessions ws_K and ws_L, where $0 \leq \mathbf{WSS}_1 \leq 1$.

$$WSS_{1KL} = \frac{\sum_{i=1}^{M} ws_K(Sig_{url_i}) \times ws_L(Sig_{url_i})}{\sqrt{\sum_{i=1}^{M} ws_K(Sig_{url_i})^2}\sqrt{\sum_{j=1}^{M} ws_L(Sig_{url_j})^2}} \quad (5)$$

The following examples illustrate the effect of this equation, using the website shown in Fig. 2.

For example, consider the two weighted sessions ws_P={(N_7,0.811,1),(N_{10},0.756,2)} and ws_Q={(N_7, 0.291,2),(N_{10},0.619,1)}, both of which are assigned 0.928 as their similarity scores. The pair has identical pages, but different significance, showing different interests of users.

Consider another example, $\mathbf{WSS}_1$ for session pair ws_K={(N_7,0.672,1),(N_9,0.521,2)} and ws_L={(N_{10},0.431, 2),(N_{11},0.542,1)}. They both are assigned the similarity value of 0 due to all pages being different.

Similarly, consider ws_K={(N_7,0.672,1),(N_9,0.521,2)} and ws_R={(N_3,0.811,1),(N_4,0.289,2)}. These are assigned similarity value of 0. If we observe the URLs more carefully, we can easily see that ws_K is actually more similar to ws_L than ws_R, if we take into consideration URL-similarity among pages (see Fig. 2). In fact, ws_K and ws_L both appear to be interested in a particular professor's profile (Fig. 2), whereas it is difficult to presume this for the pair ws_K and ws_R. Therefore, $\mathbf{WSS}_1$ clearly has some limitations in adequately representing this URL-similarity among sessions. In contrast, $\mathbf{WSS}_2$ is defined so as to overcome this limitation. It incorporates both URL-similarity and page significance. Equation (6) provides the formulation of $\mathbf{WSS}_2$ for ws_K and ws_L, where $0 < \mathbf{WSS}_2 \leq 1$.

$$WSS_{2KL} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} ws_K(Sig_{url_i}) \times ws_L(Sig_{url_j}) \times URL_{sim}(i,j)}{\sum_{i=1}^{M} ws_K(Sig_{url_i}) \times \sum_{j=1}^{M} ws_L(Sig_{url_j})} \quad (6)$$

We shall recalculate the similarity values using this equation for some of the same examples used earlier. Using $\mathbf{WSS}_2$, session pair ws_K and ws_L are assigned the similarity value of 0.80, while ws_K and ws_R are assigned the value 0.396. In general, most of the sessions contain some identical pages along with a number of different pages. Let us consider an example of such a session pair ws_E={(N_2,0.491,2),(N_7,0.845,1)} and ws_F={(N_3,0.639,2),(N_4,0.599,3),(N_7,0.825,1)}. Both sessions share identical and similar pages, but with low values for structural similarity (Fig. 2). $\mathbf{WSS}_1$ assigns 0.566 for these sessions, whereas $\mathbf{WSS}_2$ assigns 0.544, the slightly lower value is the effect of associated significance values.

As another example, consider session pairs ws_G={(N_7, 0.439,3),(N_9,0.72,1),(N_{12},0.639,2)} and ws_H={(N_7,0.819, 1),(N_{10},0.563,2)}. Both sessions have URLs with high structural similarity (Fig. 2). $\mathbf{WSS}_1$ assigns 0.342 as opposed to the value of 0.804 assigned by $\mathbf{WSS}_2$. From this we can clearly see that WS_2 takes into account page pairs with high structural similarity values much better, while WS_1 does this better for page pairs with lower structural similarity values. Note that in both cases page significance plays a critical role.

Our weighted similarity measure **WSS** utilizes these properties of both $\mathbf{WSS}_1$ and $\mathbf{WSS}_2$. It uses the maximum score of these two measures to compute a better similarity value among sessions. Equation (7) defines **WSS** for ws_K and ws_L, where $0 < \mathbf{WSS}(\mathbf{ws}_K,\mathbf{ws}_L) \leq 1$.

$$WSS(ws_K, ws_L) = \max(WSS_{1KL}, WSS_{2KL}) \quad (7)$$

WSS(ws_K,ws_L) enjoys the following three properties:

- Identity: WSS(ws_K,ws_K)=1.0.
- Symmetry: WSS(ws_K,ws_L)=WSS(ws_L,ws_K).
- Uniqueness: WSS(ws_K,ws_L)=1.0 means ws_K=ws_L.

However, in some cases **WSS** may violate Triangle Inequality: WSS(ws_K,ws_L)>WSS(ws_K,ws_M)+WSS(ws_M,ws_L).

For clustering the weighted sessions, the similarity between ws_K and ws_L is mapped to a distance measure by computing their dissimilarities, defined as follows, where $0 \leq \mathbf{WSD}(ws_K,ws_L) < 1$.

$$WSD(ws_K, ws_L) = 1 - WSS(ws_K, ws_L) \quad (8)$$

5 Usage Profile Generation

For clustering weighted sessions, we have chosen the Relational Fuzzy Subtractive Clustering (RFSC) algorithm [Suryavanshi, Shiri, and Mudur 2005] because it yields fairly accurate results, is scalable to very large datasets, is reasonably immune to noise present in web data, and is parameter independent.

Let the result of RFSC be denoted by C={C_1, C_2, ..., C_q} the set of $|q|$ fuzzy clusters, where each cluster center is an actual weighted session of the dataset, known as cluster prototype. These clusters are processed further in order to generate a set of usage profiles (or aggregate usage profiles [Mobasher et al. 2002]) so that popular pages are come forward and unpopular ones are located backward in the profile. The popularity of url_j in cluster C_z is computed by equation

(9), where 0≤Popularity[C_z,url_j]≤1.

$$Popularity[C_z, url_j] = \frac{\sum_{L=1}^{N} ws_L(Sig_{url_j}) \times MV_{C_z,L}}{|N|} \quad (9)$$

Here, $ws_L(Sig_{url_j})$ denotes the significance of url_j in ws_L and $MV_{C_z,L}$ denotes fuzzy membership value of ws_L in C_z.

6 Recommender Algorithms

In using our weighted similarity measure in recommender algorithms, an active session us_A is converted into an active weighted session ws_A by measuring page-significance [equations (1) to (3)]. Selection of nearest profiles requires similarity computation among weighted prototypes and ws_A. While using **WSS**, tiny variations in page-significance may show higher similarity for two sessions having more structurally related pages than those with more identical pages. This happens in a few cases. For avoiding such situations, we introduce a new parameter "***Overlapping Ratio***", a ratio of common URLs between ws_A and weighted prototype. Our model-based CF algorithms are given below:

- Model-based CF with Similarity and Overlapping Ratio
- Fuzzy Hybrid CF with Similarity and Overlapping Ratio

Both algorithms combine similarity (**Sim**) and overlapping ratio (**OR**) for nearest profile selection. From our various experiments, we see that this resulted in improved recommendation hits in all approaches.

6.1 Model-based CF with Sim and OR

The proposed model-based CF algorithm selects nearest cluster $C_{nearest}$ using similarity WSS(ws_A,ws_{C_z}) between weighted prototype ws_{C_z} and active weighted session ws_A, and their overlapping ratio OR_{A,C_z}. Next, it selects a set of *top_N* most popular URLs from $C_{nearest}$ and recommends this list to user. Our algorithm is described below:

Algorithm 1 Model-based CF with **Sim** and **OR**

Input: An active weighted session ws_A and a set of clusters $C = \{C_1, C_2, ..., C_q\}$. Let *NA* be the set of all URLs not in ws_A. Let $Popularity[Q, M]$ be the cluster-wise popularity matrix for all url_i.
Output: A recommendation list of *top_N* URLs.

1: For all clusters $C_z \in C$, do steps 2, 3 and 4.
2: Calculate $WSS(ws_A, ws_{C_z})$ between ws_A and ws_{C_z}.
3: Calculate OR_{A,C_z} between ws_A and ws_{C_z}.
4: Set $Combine_{A,C_z} \leftarrow WSS(ws_A, ws_{C_z}) + OR_{A,C_z}$.
5: Select $C_{nearest}$ with $\max(\forall_{C_z} Combine_{A,C_z})$.
6: For all $url_j \in$ *NA*, recommend *top_N* most popular URLs from $C_{nearest}$ using $Popularity[C_{nearest}, url_j]$.

6.2 Fuzzy Hybrid CF with Sim and OR

Our proposed fuzzy hybrid CF algorithm incorporates basic notions of both memory-based and model-based CF techniques in order to enhance accuracy and scalability of a recommender engine. In this algorithm, we divide dissimilarity range [0,1] into $|R|$ equal sub-ranges (DSR) for each cluster C_z, and distribute all extracted weighted sessions ws_L into these sub-ranges using their dissimilarity. Next, we select nearest cluster $C_{nearest}$ for ws_A with similarity WSS(ws_A, ws_{C_z}) between weighted prototype ws_{C_z} and active weighted session ws_A, and overlapping ratio OR_{A,C_z}. After computing dissimilarity WSD($ws_A, ws_{C_{nearest}}$), we select all sessions which belong to the same DSR as the one to which ws_A belongs. From this set, we select K-most nearest sessions according to their similarity to ws_A, and compute the popularity of their URLs, not accessed in ws_A. Finally, a list of *top_N* most popular URLs is recommended to the user. The algorithm is described below.

Algorithm 2 Fuzzy Hybrid CF with **Sim** and **OR**

Input: An active weighted session ws_A and a set of clusters $C = \{C_1, C_2, ..., C_q\}$. Let *NA* be the set of all URLs not in ws_A, and $DSR_{C_z,r} \in |R|$ be the dissimilarity sub-ranges of C_z.
Output: A recommendation list of *top_N* URLs.

1: For all clusters $C_z \in C$, do steps 2, 3 and 4.
2: Calculate $WSS(ws_A, ws_{C_z})$ between ws_A and ws_{C_z}.
3: Calculate OR_{A,C_z} between ws_A and ws_{C_z}.
4: Set $Combine_{A,C_z} \leftarrow WSS(ws_A, ws_{C_z}) + OR_{A,C_z}$.
5: Select $C_{nearest}$ with $\max(\forall_{C_z} Combine_{A,C_z})$.
6: Set $WSD(ws_A, ws_{C_z}) \leftarrow 1 - WSS(ws_A, ws_{C_z})$.
7: Select $DSR_{C_{nearest},r}$ by using $WSD(ws_A, ws_{C_z})$.
8: Choose $ws_{neighbor}$ belong to $DSR_{C_{nearest},r}$.
9: Select $ws_{K_{nearest}}$ using WSS(ws_A,$ws_{neighbor}$).
10: For all $url_j \in$ *NA* do step 11
11: For each $ws_{K_{nearest}}$ do step 12
12: If $url_j \in ws_{K_{nearest}}$, then do step 13
13: Set $popularity(url_j) \leftarrow WSS(ws_A, ws_{K_{nearest}}) \times ws_{K_{nearest}}(Sig_{url_j})$.
14: Using $popularity(url_j)$, recommend *top_N* most popular URLs.

7 Experiments and Results

We carried out a series of experiments to evaluate the efficiency and effectiveness of the proposed weighted similarity measure (*WSS*). To compare recommendation performance, we also carried out the same experiments with four other similarity measures, namely, Pearson correlation coefficient (*PCC*), Jaccard coefficient (*JC*), Cosine similarity (*CS*), and the measure proposed in [Nasraoui et al. 1999], which we shall call as Binary Session Similarity (*BSS*). All experiments were performed on an Intel(R) Xeon(R) 3400 series based workstation running at 2.67 GHz with 4GB RAM. For the experiments, we used access log data from the web server of our Computer Science department during December 31, 2004 to January 15, 2005. After data clean-

Table 1: Clusters for training set of 10864 weighted sessions.

Usage Profiles	UP_1	UP_2	UP_3	UP_4	UP_5
No. of Clusters	37	33	16	36	68

Table 2: Comparison of results when using model-based CF algorithm for most-significant hidden set.

Model-based CF Algorithm															
	UP_1			UP_2			UP_3			UP_4			UP_5		
top_N	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%
5	26.3	5.3	19.2	19.7	3.9	10.6	19.6	3.9	10.1	26.6	5.3	17.3	**36.5**	**7.3**	**25.7**
10	32.4	3.2	20.0	29.1	3.0	11.9	27.5	2.8	11.0	32.7	3.3	18.1	**44.4**	**4.4**	**26.8**
15	36.2	2.4	20.3	34.1	2.3	12.3	33.1	2.2	11.5	36.9	2.5	18.4	**48.9**	**3.3**	**27.1**
20	38.7	1.9	20.5	38.0	1.9	12.5	36.0	1.8	11.6	39.3	2.0	18.5	**51.8**	**2.6**	**27.3**

Table 3: Comparison of results when using fuzzy hybrid CF algorithm for most-significant hidden set.

Fuzzy Hybrid CF Algorithm															
	UP_1			UP_2			UP_3			UP_4			UP_5		
top_N	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%
5	34.7	6.4	24.9	31.4	6.3	22.3	28.2	5.7	19.3	28.6	5.7	19.7	**37.1**	**7.4**	**27.9**
10	41.3	4.1	25.8	38.6	3.9	23.4	35.0	3.5	20.2	34.2	3.4	20.5	**44.6**	**4.5**	**28.9**
15	44.6	3.0	26.1	42.6	2.8	23.6	38.4	2.6	20.4	37.4	2.5	20.7	**51.3**	**3.4**	**29.3**
20	47.0	2.4	26.2	45.4	2.3	23.7	40.9	2.1	20.6	39.4	2.0	20.9	**53.9**	**2.7**	**29.4**

ing, we had about 46 MB of 2,00,000 cleaned records. After session extraction, we got 16,816 usage sessions and 12,685 distinct URLs. After removing sessions with length of 1 or 2, and computing page-significance, we had 13,580 weighted sessions with average session length of 7.35 and over 99% of data sparsity, defined as [1 - $\frac{Nonzero\ Entries}{Total\ Entries}$]. We randomly divided the dataset into a training set and a test set using different ratios (90%:10%, 80%:20%, 70%:30%, and 60%:40%). For each test session, some pages are hidden, forming a *Hidden set*. Below we show the results for the 80%:20% case. Other results are similar, with decrease in quality as test set size increases. Let *top_N* denote the set of recommended pages. If a hidden page is present in recommendation list, we call it a *hit*. From the training dataset, we generated the following usage profiles using five different similarity measures and the fuzzy clustering algorithm.

- UP_1=$\{up_{11}, up_{12}, ..., up_{1L}\}$ using *BSS* measure.
- UP_2=$\{up_{21}, up_{22}, ..., up_{2R}\}$ using *JC* measure.
- UP_3=$\{up_{31}, up_{32}, ..., up_{3X}\}$ using *PC* measure.
- UP_4=$\{up_{41}, up_{42}, ..., up_{4Y}\}$ using *CS* measure.
- UP_5=$\{up_{51}, up_{52}, ..., up_{5Q}\}$ using proposed *WSS* measure.

In our experiments, the recommender approach utilizing usage profiles UP_1 and *BSS* measure is called as the $\mathbf{UP}_1$ approach. Similarly, for other cases listed above we call them as $\mathbf{UP}_2$, $\mathbf{UP}_3$, $\mathbf{UP}_4$, and $\mathbf{UP}_5$ approaches, respectively.

Let *NP* denote the number of nearest clusters, and *Nearest_K* denote the K-nearest neighbors of each test session. Let *DSR* denote the dissimilarity sub-range. We show the results of experiments by keeping *NP* constant at 1, *Nearest_K* at 100, *DSR* at 0.10, and varying *top_N* to 5, 10, 15, and 20, respectively. Experiments for higher values of *NP* show that our method improves further while the others show further deterioration.

7.1 Performance Evaluation Metrics

We used the metrics, recall, precision, and mean reciprocal hit-rank to evaluate effectiveness. For efficiency, we used the recommendation time (in seconds) per user.

1. *Recall(R)*: It is the ratio of hits in *Hidden set*. Higher recall value means improved performance. *Recall*(%) is defined as follows.

$$Recall(\%) = \frac{|Hidden\_set \cap top\_N|}{|Hiddent\_set|}(\%) \qquad (10)$$

2. *Precision(P)*: It shows the ability of recommender system to give accurate recommendations. Larger value for precision leads to better performance. *Precision*(%) is defined as follows.

$$Precision(\%) = \frac{|Hidden\_set \cap top\_N|}{|top\_N|}(\%) \qquad (11)$$

3. *Mean Reciprocal Hit-Rank(MRHR)*: It assesses the recommendation quality. Earlier occurred *hits* in *top_N* are given more weight in *MRHR*. Higher the *MRHR*, better the recommendation quality. Let H be the number of hits, positioned at p_1, p_2,...,p_H in *top_N*. *MRHR*(%) is defined as follows.

$$MRHR(\%) = \frac{1}{|Test\_set|}\sum_{i=1}^{H}\frac{1}{p_i}(\%) \qquad (12)$$

7.2 Performance Analysis

Table 1 shows the total number of clusters obtained from using the five different session similarity measures. We believe that the higher number of clusters resulting from the use of our weighted measure is due its better discrimination capabilities. We conducted experiments based on two separate cases as follows.

Performance of Most Significant Hidden Set

We hide the most significant (i.e., rank 1) page from each test session. Tables 2 and 3 present the recommendation results in terms of *Recall*, *Precision*, and *MRHR* for the new model-based and the new fuzzy hybrid CF approaches respectively. From these tables, we can see that our $\mathbf{UP}_5$ (which uses the *WSS* measure) outperforms the other approaches, by providing recommendations with higher recall, better precision, and greater MRHR, with only a minor increase in recommendation time. This is shown in Tables 7 and 8.

Table 4: Comparison of results when using model-based CF algorithm for randomly selected hidden set.

Model-based CF Algorithm															
	UP_1			UP_2			UP_3			UP_4			UP_5		
top_N	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%
5	28.4	5.7	22.1	16.0	3.2	8.7	10.1	2.0	5.2	24.6	4.9	15.1	**32.1**	**6.4**	**22.3**
10	35.4	3.5	23.0	21.3	2.1	9.4	13.0	1.3	5.6	30.2	3.0	15.8	**38.2**	**3.8**	**23.3**
15	38.4	2.6	23.3	25.1	1.7	9.7	16.2	1.1	5.8	34.5	2.3	16.1	**42.5**	**2.8**	**23.5**
20	41.5	2.1	23.5	28.4	1.4	9.9	18.7	0.9	6.0	36.9	1.9	16.3	**45.5**	**2.3**	**23.7**

Table 5: Comparison of results when using fuzzy hybrid CF algorithm for randomly selected hidden set.

Fuzzy Hybrid CF Algorithm															
	UP_1			UP_2			UP_3			UP_4			UP_5		
top_N	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%
5	34.5	6.9	25.8	25.0	5.0	17.7	24.6	4.9	15.8	31.7	6.3	21.5	**37.6**	**7.5**	**26.4**
10	41.4	4.1	26.9	31.8	3.2	18.6	31.8	3.2	16.7	37.9	3.8	22.3	**45.8**	**4.6**	**27.3**
15	43.5	2.9	27.2	35.2	2.4	18.9	34.6	2.3	17.0	40.9	2.7	22.6	**50.2**	**3.4**	**27.5**
20	46.4	2.3	27.4	38.3	1.9	19.0	36.6	1.8	17.1	43.6	2.2	22.7	**53.3**	**2.7**	**27.6**

Table 6: Comparison of results when using model-based CF algorithm for random *high-significant* pages.

Model-based CF Algorithm															
	UP_1			UP_2			UP_3			UP_4			UP_5		
top_N	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%	R%	P%	MRHR%
5	26.9	5.4	21.2	20.9	5.2	10.8	12.6	2.5	6.4	29.6	5.9	19.0	**38.0**	**7.6**	**26.3**
10	32.8	3.3	22.0	27.6	2.8	11.7	16.2	1.6	6.9	34.3	3.4	19.7	**45.1**	**4.5**	**27.3**
15	35.5	2.4	22.2	32.6	2.2	12.1	20.2	1.4	7.2	38.2	2.5	20.0	**49.7**	**3.3**	**27.6**
20	38.9	1.9	22.4	36.4	1.8	12.3	24.2	1.2	7.4	40.7	2.0	20.1	**51.6**	**2.6**	**27.7**

Table 7: Recommendation time (in seconds) per user for model-based CF algorithm with *top-N*=20.

top_N	UP_1	UP_2	UP_3	UP_4	UP_5
20	0.68	0.65	**0.40**	0.55	0.75

Table 8: Recommendation time (in seconds) per user for fuzzy hybrid CF algorithm with *top-N*=20.

top_N	UP_1	UP_2	UP_3	UP_4	UP_5
20	**1.61**	2.10	4.91	2.13	2.0

Performance of Randomly Selected Hidden Set

We hide randomly a page from each test session. Tables 4 and 5 present the overall performance in terms of *Recall*, *Precision*, and *MRHR* for the new model-based and the new fuzzy hybrid CF approaches. From these tables, we observe that our $\mathbf{UP}_5$ provides a better recommendation quality with respect to others for randomly selected hidden set.

We divide the page-significance range $[0, 1]$ into three different ranges including "High significance" range from 0.41 to 1.0, "Mid significance" range from 0.11 to 0.4, and "Low significance" range from 0.0 to 0.1. Out of randomly hidden 2,716 pages, a total of 1,122 is identified as *high-significant* pages. Our goal is to determine how well *high-significant* pages are recommended, as ideally, they should not be missed by any recommender system. Table 6 shows the recommendations for random hidden *high-significant* pages for the new model-based CF approach. From these numbers, we find that our $\mathbf{UP}_5$ gives high quality recommendations with enhanced recall, improved precision, and larger MRHR as compared to others. The same trend is seen for the new fuzzy hybrid CF approach in Figs. 3 and 4, and in Table 9. The computation time for this hidden set is the same as that for the most significant hidden set.

Table 9: Comparison of MRHR(%) when using fuzzy hybrid CF algorithm for random *high-significant* pages.

top_N	UP_1	UP_2	UP_3	UP_4	UP_5
5	28.2	24.0	20.6	20.6	**32.3**
10	29.1	25.0	21.4	21.4	**33.3**
15	29.4	25.2	21.7	21.7	**33.6**
20	29.5	25.4	21.8	21.8	**33.7**

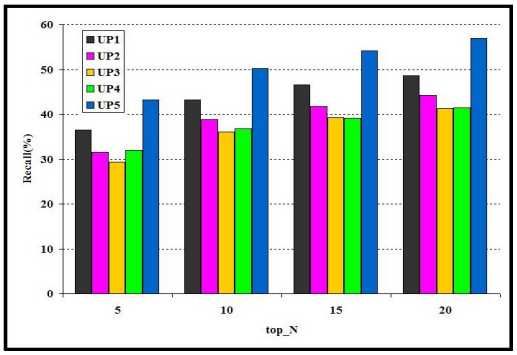


Figure 3: Comparison of Recall(%) when using fuzzy hybrid CF algorithm for random *high-significant* pages.

8 Conclusion and Future Work

Effective user grouping by clustering web usage data can lead to usage profiles that improve performance of recommender systems. Successful clustering, however, depends on

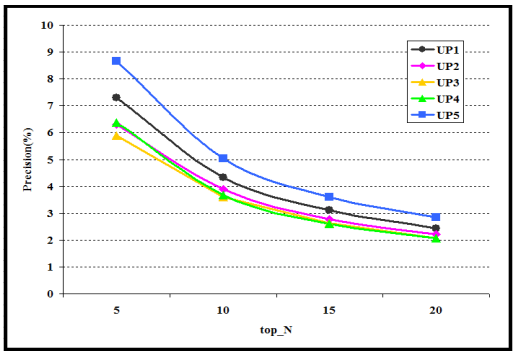


Figure 4: Comparison of Precision(%) when using fuzzy hybrid CF algorithm for random *high-significant* pages.

how well user interests are captured and accommodated by the similarity measure that is used. In this paper, we have proposed a weighted session similarity measure to assess usage session similarity by considering both page significance and URL structure similarity. Two model-based CF algorithms are adapted to use this measure for evaluation of its effectiveness. Numerous experiments confirm that our similarity measure helps discover effective usage profiles from large web log data. This is demonstrated by using these usage profiles into a recommender system. Our experiments include performance comparison with four other popular similarity measures. Our weighted session similarity measure clearly outperforms others by providing recommendations of higher quality. In the immediate future, we intend to extend our similarity measure by incorporating semantics [Diligenti, Gori, and Maggini 2011].

Acknowledgments

This work was supported in part by Natural Sciences and Engineering Research Council (NSERC) of Canada, and by Concordia University.

References

Bose, A.; Beemanapalli, K.; Srivastava, J.; and Sahar, S. 2006. Incorporating concept hierarchies into usage mining based recommendations. In *Proceedings of 8th Knowledge discovery on the web int'l conference on Advances in web mining and web usage analysis (WebKDD'06)*, 110–126.

Castellano, G.; Fanelli, A. M.; Mencar, C.; and Torsello, M. A. 2007. Similarity-based fuzzy clustering for user profiling. In *Proceedings of Workshop on 2007 IEEE/WIC/ACM Conferences on Web Intelligent Agent Technology*, 75–78.

Castellano, G.; Castiello, C.; Dell'Agnello, D.; Fanelli, A. M.; Mencar, C.; and Torsello, M. A. 2010. Learning fuzzy user profiles for resource recommendation. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 18(4):389–410.

Chan, P. K. 1999. A non-invasive learning approach to building web user profiles. In *Workshop on Web usage analysis and user profiling , Fifth International Conference on Knowledge Discovery and Data Mining*, 7–12.

Deshpande, M., and Karypis, G. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems* 22(1):143–177.

Diligenti, M.; Gori, M.; and Maggini, M. 2011. A unified representation of web logs for mining applications. *Information Retrieval* 14(3):215–236.

Li, C., and Lu, Y. 2007. Similarity measurement of web sessions by sequence alignment. In *Proceedings of the Workshop on 2007 IFIP International Conference on Network and Parallel Computing*, 716–720.

Liu, H., and Keselj, V. 2007. Combined mining of web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data and Knowledge Engineering* 61(2):304–330.

Mobasher, B.; Dai, H.; Luo, T.; and Nakagawa, M. 2002. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery* 6(1):61–81.

Nasraoui, O.; Frigui, H.; Joshi, A.; and Krishnapuram, R. 1999. Mining web access logs using relational competitive fuzzy clustering. In *Proceedings of 8th International Fuzzy Systems Association World Congress(IFSA'99)*, 230–237.

Pierrakos, D.; Paliouras, G.; Papatheodorou, C.; and Spyropoulos, C. D. 2003. Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction* 13(4):311–372.

Shahabi, C.; Zarkesh, A.; Adibi, J.; and Shah, V. 1997. Knowledge discovery from usersweb-page navigation. In *Proceedings of 7th International Workshop on Research Issues in Data Engineering(RIDE'97) High Performance Database Management for Large-Scale Application*, 20–29.

Spiliopoulou, M.; Mobasher, B.; Berendt, B.; and Nakagawa, M. 2003. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *INFORMS Journal of Computing* 15(2):171–190.

Suryavanshi, B. S.; Shiri, N.; and Mudur, S. P. 2005. An efficient technique for mining usage profiles using relational fuzzy subtractive clustering. In *Proceedings of International Workshop on Challenges in Web Information Retrival and Integration(WIRI'05)*, 23–29.

Wang, S.; Xu, C.; and Wu, R. 2008. Cluster method based on fuzzy multisets for web pages and customer segments. In *Proceedings of 2008 International Seminar on Business and Information Management*, 125–128.

Wang, J.; Zhang, N. Y.; and Yin, J. 2010. Collaborative filtering recommendation based on fuzzy clustering of user preferences. In *Proceedings of Seventh International Conference on Fuzzy Systems and Knowledge Discovery(FSKD 2010)*, 1946–1950.

Xiao, J.; Zhang, Y.; Jia, X.; and Li, T. 2001. Measuring similarity of interests for clustering web-users. In *Proceedings of 12th Australasian database conference*, 107–113.

Xue, G. R.; Lin, C.; Yang, Q.; Xi, W.; Zeng, H. J.; and Yu, Y. 2005. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of ACM SIGIR Conference(SIGIR'05)*, 1147–121.