Similarity-Based Clustering of Web Transactions

Giuseppe Manco[1], Riccardo Ortale[2], and Domenico Saccà[1,2]

[1] ICAR-CNR
Via Bucci 41c
I87036 Rende (CS) - Italy
Ph: +39 0984 831728 Fax: +39 0984 839054
e-mail: manco@isi.cs.cnr.it

[2] DEIS, University of Calabria
Via Bucci 41c
I87036 Rende (CS) - Italy
Ph: +39 0984 494750 Fax: +39 0984 839054
e-mail: ortale@si.deis.unical.it, sacca@unical.it

Abstract. We propose a measure to compute similarity between sequences containing accesses to Web pages, and a centroid-based clustering approach for grouping sessions of accesses to a Web site. The notion of sequence similarity is parametric to *(i)* the sequence topology, and *(ii)* the similarity among Web pages within the sequences. In our formalization, two Web pages are similar if they can be considered synonymies not only from a content point of view, but also from a usage point of view, i.e., if users exhibit the same behavior on both pages. In order to design the clustering scheme, the notion of cluster centroid is further investigated. In our formalization, a centroid is formalized as a generalized medoid, i.e., as a sequential pattern that appears frequently in the cluster under investigation. The advantage of such a definition is envisaged in the application of the clustering technique to the personalization of Web experience.

Keywords: Similarity Measures, Generalized Medoids, Sequential Patterns, Cluster profiles.

1 Introduction

Analysis of user access patterns (extracted from Web server logs) makes it possible to automatically learn profiles for groups of users having similar navigation behavior. Each profile describes the information interests, preferences and requirements of a given subset of users. Profile information can be then exploited to detect current users' browsing goals based on their recent navigation history and, consequently, support their navigation by suggesting them pages potentially interesting. Personalization effectiveness heavily relies on user profile reliability which, in turn, depends on the accuracy with which user navigation behavior is modelled. In order to increase profile reliability, it is necessary to reconstruct not only the visited pages, but also the chronological order with which they have been accessed. This approach allows for an exact reconstruction of user navigation behavior and, consequently, of their requirements and preferences.

However, most of the approaches to Web personalization do not address the sequential nature of user browsing. Usually, user sessions are modelled by means of n-dimensional vectors defined over the space of Web pages within a given Web site: vector dimensions correspond to specific Web pages. Depending on the nature of the values associated to these dimensions, different kinds of limited user behavior analysis can be performed. Binary vectors simply indicate whether or not a given page has been accessed. This is the poorest kind of behavior analysis, since it is not possible to either count how many times a given page has been visited or distinguish which pages, among the visited ones, are effectively of interest to users: all visited pages appear as equally important. In case of non binary values, it is possible to advise ad-hoc significance weights which take into consideration a number of page parameters, such as the viewing time and the access frequency. However, any weighting criteria adopted with a vector representation cannot take into consideration essential, navigation-intrinsic information: the order with which pages have been accessed. This eventually affects the accuracy of the profile learning phase. In fact, users who have accessed a number of identical pages may be assumed to share the same browsing goals, without actually taking into consideration that different chronological sequences of page accesses, though regarding the same Web pages, may reflect distinct navigation purposes and requirements. User sessions are better modelled as sequences of pages accesses, which can be effectively used to model a real browsing scenario. Sequences in fact take into consideration the order with which Web pages have been accessed.

An important source of information about user navigation behavior is represented by page contexts, i.e. subsets of pages visited immediately before and after a given Web page. Page contexts can be exploited to provide explanations about patterns occurring within each user session. To this purpose, sequences are useful not only because they allow to detect when specific information

requirements arise, but also because they may be leveraged to explicitly devise suitable weighting criteria for the significance of pages within each session based on their access order. This allows to accurately detect the different preferences, requirements and goals of users who have accessed similar pages but with different purposes. Sequences, in fact, allow to compare navigation actions of such users not only in terms of visited pages, but also in terms of the topology of their navigation paths.

This paper introduces an approach to the delivery of personalized recommendations which exploits sequences as a model of user sessions/transactions. The proposed approach is based on two novel similarity measures, which are both presented and discussed:

- a measure for computing similarity between Web pages, which detects page content and usage synonymies;
- and a measure to evaluate similarity between user sessions which is parametric to the topology of sequences themselves.

Personalized recommendations are delivered exploiting user profiles learnt from clusters of similar user sessions. Since traditional vector-based methodologies to profile building cannot be applied in the case of sequences, a new approach to the detection of usage profiles is discussed. Moreover, the notion of cluster centroid is investigated. In our formalization, a centroid is formalized as a generalized medoid, i.e., as a sequential pattern that appears frequently in the cluster under investigation.

2 Problem Statement

Assume $\mathcal{U} = \{p_1, p_2, \ldots, p_n\}$ is the set of all pages within a given Web site (p_i corresponds to the *i-th* Web page). Any user transaction tr is a time ordered sequence of Web pages in $\mathcal{U}$ and therefore can be formally modelled as a sequence of pairs, such that each pair is made up of a certain visited page and the viewing time spent on that page: $tr = \{(p_{i_1}, t_1), (p_{i_2}, t_2), \ldots (p_{i_m}, t_m)\}$. Length of sequences changes with respect to the particular user transaction considered. Given a generic transaction sequence tr, two more sequences can always be extracted from tr:

- a Web page sequence $wps_{tr} = \langle p_{i_1}, p_{i_2}, \ldots, p_{i_m} \rangle$ which consists of the page accesses within tr;
- a viewing time sequence $vts_{tr} = \langle t_1, t_2, \ldots, t_m \rangle$, where an element with position j is the viewing time spent on the Web page at the same position within wps_{tr}.

Example 1. As a toy example, we can consider a web site containing 4 web pages (resp. p_1, p_2, p_3, p_4) and 10 user navigation sessions, described below in terms of sequences of page accesses:

$$wps_{s_1} : \langle p_1, p_4 \rangle \qquad wps_{s_6} : \langle p_1, p_3 \rangle$$
$$wps_{s_2} : \langle p_1, p_3, p_2 \rangle \qquad wps_{s_7} : \langle p_1, p_4, p_2 \rangle$$
$$wps_{s_3} : \langle p_1, p_2, p_4 \rangle \qquad wps_{s_8} : \langle p_3, p_2, p_1 \rangle$$
$$wps_{s_4} : \langle p_2, p_3 \rangle \qquad wps_{s_9} : \langle p_4, p_3 \rangle$$
$$wps_{s_5} : \langle p_3, p_4 \rangle \qquad wps_{s_{10}} : \langle p_2, p_4 \rangle$$

Each such sequence has a corresponding time sequence, describing the viewing time exhibited by each page within the session. For example, we have $vts_{s_2} = \langle 8\ 3\ 10 \rangle$ and $vts_{s_8} = \langle 5\ 6\ 7 \rangle$. □

The goal is to support user navigation throughout a Web site since the earliest stages of their browsing activity. Formally, given a user u and her/his recent navigation history rh (a subsequence consisting of the last n visited pages within the original navigation session of u), assume that u clicks on a link to a page $p_i \in \mathcal{U}$. In order to proactively detect requirements and preferences of u, a set $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ of clusters of users with similar transactions is exploited. Precisely, a set $\mathcal{P} = \{m_{\mathcal{C}_1}, \ldots, m_{\mathcal{C}_k}\}$ of user profiles is generated such that any profile $m_{\mathcal{C}_i}$ is assigned to a corresponding cluster $\mathcal{C}_i$. $m_{\mathcal{C}_i}$ is computed as a *generalized* medoid of $\mathcal{C}_i$. It is originally a particular sequence within $\mathcal{C}_i$ which is the most similar to the other sequences in the same cluster. Consequently that sequence is generalized to synthetically describe the typical navigational behavior of all users within cluster $\mathcal{C}_i$. By evaluating session similarity between rh and all the above user profiles, it is possible to find a subset, *ClosestProfiles*, of those profiles which best reflect the navigation behavior of u. Profiles in *ClosestProfiles* are eventually used to dynamically generate recommendations of interest for user u: such recommendations are added to the contents of page p_i.

According to the above formalization, the problem of providing support to personalization can be decomposed in the following:

- Definition of preprocessing techniques for extracting relevant features from the data. Such techniques are mainly useful for sessioning the accesses of each user in sequences, and in detecting the contents of each page.
- Definition of a notion of similarity between the identified sequences.
- Definition of a clustering-based methodology for identifying profiles.
- Definition of criteria for matching profiles to current on-line user behaviors.

We now analyze each aspect in turn.

3 Data preprocessing

Data preprocessing consists of two phases required to convert raw usage and content data into a minable data set. Such phases are usage data preprocessing and content data preprocessing.

Usage data preprocessing. A number of tasks are carried out at this stage on the usage data extracted from Web server logs: data cleaning, user identification, user session reconstruction, identification of pageviews (henceforth simply referred to as pages), path completion, transaction identification. These tasks are not detailed here: an explanation of their usefulness together with a detailed description of each of them can be found in [5, 4, 6]. Here, it is assumed that the portion of the minable data set containing usage data is a collection of TN page sequences corresponding to as many user transactions. An optional preprocessing phase, support filtering [21, 3] can be finally used in order to eliminate noise from usage data by removing all those transaction page accesses having either very low or extremely high support. These page accesses cannot be profitably leveraged to characterize the behavior of any group of users.

Content data preprocessing. Content data preprocessing consists of traditional information retrieval techniques. First, each Web page is parsed in order to extract the contained words. Therefore a list of the words extracted from all pages is obtained. Two further activities are carried out on this list: the removal of any word belonging to a suitable stop list (such as, e.g., the one devised in [7]), and the reduction of the remaining words to their stems. A site dictionary SD is build as a collection of unique word stems. SD represents a space of features (words), which can be modelled as a feature vector of size $|SD|$. Finally, every Web page $p_i \in \mathcal{U}$ is assigned its own feature vector fv_{p_i}, whose dimensions are the weights of the corresponding features within p_i. These weights are computed by leveraging the traditional *tfidf* technique [26].

4 Evaluating Similarity of Web Sessions

The problem of computing similarities among user navigation paths through a Web site is faced by combining the computation of *Web page similarity* and *sequence similarity*.

To our purposes, a user session can be considered as a sequence of pages visited by the same user. As a consequence, a technique for efficiently evaluating the similarity of time sequences of discrete values can be devised. However, in the context of Web pages, a major improvement w.r.t. such approaches can be considered. Typically, definitions of session similarity measures are based on looking for identical pages within two different user sessions: the higher the number of common pages is, the more similar such sessions are considered. As a consequence, the approach proposed

for evaluating Web user session similarity can be made dependent on a particular definition of similarity between Web pages. Indeed, any two different user sessions having no matching pages can be still considered similar based on similarity of either contents or usage of their pages.

4.1 Similarity between Web Pages

In general, two Web pages should be considered similar if both correspond to the same page, or, otherwise, if they are somehow related to each other through either content or usage. This intuition is formalized next. Given any two Web pages p_i,$p_j \in \mathcal{U}$, a page similarity measure can be defined as

$$sim^{pg}(p_i, p_j) = \begin{cases} 1 & \text{if } i = j \\ \alpha_1 sim_{content}(fv_{p_i}, fv_{p_j}) + \alpha_2 sim_{usage}(p_i, p_j) & \text{if } i \neq j \end{cases} \quad (1)$$

where $sim_{content}$ refers to content similarity, and sim_{usage} to usage similarity. Values α_1 and α_2 (chosen in a way such that $\alpha_1 + \alpha_2 = 1$) are leveraged to explicitly quantify how both content and usage similarity affect the resulting page similarity measure.

The term $sim_{content}(fv_{p_i}, fv_{p_j})$ computes the cosine similarity between the feature vectors associated to the pages p_i and p_j:

$$sim_{content}(fv_{p_i}, fv_{p_j}) = \frac{\sum_{t=1}^{|SD|} fv_{p_i}[t] * fv_{p_j}[t]}{|fv_{p_i}||fv_{p_j}|} \quad (2)$$

Usage similarity, $sim_{usage}(p_i, p_j)$, can be expressed by the following formula

$$sim_{usage}(p_i, p_j) = \sum_{l=1}^{\infty} w^l c_{ij}^{(l)} \quad (3)$$

Terms $c_{ij}^{(l)}$ are necessary to compute page similarity from a usage point of view. To this purpose, the topology of accesses to a Web site can be modelled as a directed labelled graph $G = \langle N, A \rangle$, where $N = \{i | p_i \in \mathcal{U}\}$ is a set of nodes corresponding to Web pages and A is a set of links connecting the above pages. In A, an arc a_{ij} connects two nodes p_i and p_j if and only if there exists a session s such that both $p_i \in s$ and $p_j \in s$. Each arc a_{ij} is labelled by the set of all the sessions containing both p_i and p_j. Figure 1 shows the resulting graph of example 1.

The intuition around the graph representation is the following. Any two web pages p_i and p_j share some degree of usage similarity if there exists at least a path connecting their corresponding nodes within the dataset of user sessions. Navigation paths from p_i to p_j and from p_j to p_i are both considered as evidence of usage similarity between the two pages. More precisely, the degree of usage similarity for a pair of pages is parametric to:

- the distance they exhibit within a session;

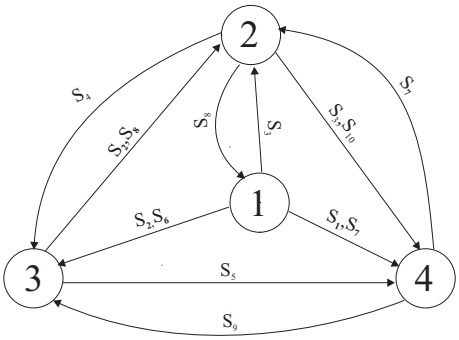

Fig. 1. Example page graph

– the number of sessions supporting both the pages.

As a consequence, an element $c_{ij}^{(l)}$ can be defined by suitably considering all those user sessions in which pages p_i and p_j are accessed together (disregarding their order of appearance) and have a link distance (i.e., number of intermediate arcs) l from each other.

In equation (3) usage similarity between any two Web pages is conceptually defined as the sum of the contributes due to all paths, between the corresponding graph nodes, with any length. In practice, a reasonable approximation for the maximum path length may be the mean length of user sequences within the session dataset.

Figure 2 describes an approach for computing $c^{(l)}$. Notice how in step 3 of the proposed algorithm the similarity of pages p_i and p_j is weighted by the discrepancy between the weight of the arc from p_i to p_j and that from p_j and p_i. This step is necessary to ensure symmetry in $c^{(l)}$, and contemporarily to catching the semantic difference, from a usage point of view, between p_i and p_j. The computation of c^l, exemplified in the following example, is quite simple and can benefit from suitable memory organizations of the data.

Example 2. Matrices $c^{(l)}$ are computed on the site graph of figure 1. In particular, since the mean length of sequences in example 1 is 2, we can compute only matrices $c^{(1)}$ and $c^{(2)}$.

$$c^{(1)} = \begin{bmatrix} 0 & \frac{2}{10} & \frac{1}{6} & \frac{1}{6} \\ \frac{2}{10} & 0 & \frac{4}{10} & \frac{2}{10} \\ \frac{1}{6} & \frac{4}{10} & 0 & \frac{2}{10} \\ \frac{1}{6} & \frac{2}{10} & \frac{2}{10} & 0 \end{bmatrix} \qquad c^{(2)} = \begin{bmatrix} 0 & \frac{1}{6} & \frac{1}{11} & \frac{1}{11} \\ \frac{1}{6} & 0 & 0 & 0 \\ \frac{1}{11} & 0 & 0 & 0 \\ \frac{1}{11} & 0 & 0 & 0 \end{bmatrix}$$

and $c^{(l)} = \mathbf{0}$ for $l > 2$. □

In equation (3), w^l is a monotone decreasing function, used to suitably weight the significance of the paths between Web pages based on the length of these paths. Precisely, paths between two

Input: A set $\mathcal{S} = \{s_1, \ldots, s_N\}$ of sessions and an integer $l \leq n$.

Output: usage similarity matrix $c^{(l)}$.

Method: Perform the following steps:

1. initially, set $c_{ij}^{(l)} := 0$ for each i, j.
2. for each session $s \in \mathcal{S}$
   - for each pair of items $i, j \in s$, such that i and j have distance l within s, do $c_{ij}^{(l)} = c_{ij}^{(l)} + \frac{1}{N}$
3. for each pair of items $i, j \in s$ recompute new values for $c_{ij}^{(l)}$ according to the following:

$$c_{ij}^{(l)} := \frac{c_{ij}^{(l)} + c_{ji}^{(l)}}{1 + |c_{ij}^{(l)} - c_{ji}^{(l)}|}$$

4. return $c^{(l)}$

Fig. 2. Computation of $c^{(l)}$

Web pages p_i and p_j with length l are considered as more indicative of usage correlation than those paths with length $l' > l$. Some interesting definitions of w^l are either $w^l = \frac{1}{l}$ (linear decreasing), $w^l = \frac{1}{l^2}$ (quadratic decreasing) or $w^l = \frac{1}{2^l}$ (exponential decreasing).

Example 3. Assuming $w^l = \frac{1}{2^l}$ and $sim_{content}(p_i, p_j) = 0$ for each $p_i, p_j \in \mathcal{U}$, we can exploit the matrices $c^{(l)}$ computed in example 2 and finally obtain the following similarity matrix:

$$\begin{bmatrix} 0 & \frac{17}{120} & \frac{14}{132} & \frac{14}{132} \\ \frac{17}{120} & 0 & \frac{2}{10} & \frac{1}{10} \\ \frac{14}{132} & \frac{2}{10} & 0 & \frac{1}{10} \\ \frac{14}{132} & \frac{1}{10} & \frac{1}{10} & 0 \end{bmatrix}$$

For example, if $\alpha_2 = \frac{1}{2}$, the value $sim^{pg}(p_1, p_4)$ can be computed as follows:

$$sim^{pg}(p_1, p_4) \equiv \alpha_2 sim_{usage}(p_1, p_4) = 0.0530$$

□

It is worth noticing how, starting from the above defined similarity between adjacent nodes in the site graph, the approach can be easily extended to compute a more refined notion of similarity between any two connected components in the graph. We plan to implement such refinements in a future extension of the paper.

4.2 Similarity between Web User Sessions

Similarity between any two generic user sessions can be built upon inner page similarity and intrinsic session information regarding page contexts within the corresponding session sequences.

To this purpose, we can exploit an approach based on the model of *time warping distance*. A time warping distance conceptually measures how similar two sequences are, not only in terms of visited pages, but also taking into consideration other essential information such as:

- similarity between the viewing times associated to the compared pages within the two sessions;
- and the topological similarity between such Web sessions.

The above information can be formalized by weighting session similarity on a *per-page* basis. Therefore, any two Web sessions are similar if they contain similar pages, which are viewed for a similar time extent and are accessed within similar page contexts.

Let us consider two Web user sessions $s_1 = \{(p_1, t_1^1), \ldots, (p_m, t_m^1)\}$ and $s_2 = \{(q_1, t_1^2), \ldots, (q_n, t_n^2)\}$. The similarity between s_1 and s_1 can be defined as:

$$sim(s_1, s_2) = sim^{1,1}(s_1, s_2)$$

where $sim^{i,j}(s_1, s_2)$ represents the similarity between the fragments $\{(p_i, t_i^1), \ldots, (p_m, t_m^1)\}$ and $\{(q_j, t_j^2), \ldots, (q_n, t_n^2)\}$ of s_1 and s_2. In particular,

- if either $i = m + 1$ or $j = n + 1$ (that is, we are considering an empty sequence), then

$$sim^{i,j}(s_1, s_2) = 0$$

- otherwise, the similarity can be computed by considering the cointribution of analysing the similarity of the head of each sequence, and the maximum similarity in the remaining subsequences:

$$sim^{i,j}(s_1, s_2) = \frac{sim^{pg}(p_i, q_j)}{1 + \alpha_{ij}} + \max\left\{sim^{i,j+1}(s_1, s_2), sim^{i+1,j}(s_1, s_2), sim^{i+1,j+1}(s_1, s_2)\right\}$$

  In the above equation, the first term computes the contribution of the similarity of p_i and q_j. To this purpose, we consider the page similarity, by weighting such a similarity with a factor α_{ij}, directly computable by both the context in which p_i and q_j appear and their associated viewing times. A naive definition, which does not exploits page context information, is $\alpha_{ij} = |t_i^1 - t_j^2|$. However, it is possible to refine such a definition by, e.g., taking into account the page similarity between the pages directly preceding (resp. following) p_i and q_j in s_1 and s_2.

It is easy to see that the above definition of $sim^{i,j}$ is well-founded. Moreover, the computation of $sim(s_1, s_2)$ can be accomplished in $O(mnc_p)$ by exploiting a dynamic-programming approach. Here, c_p is the cost of computing the similarity between Web pages.

Example 4. Let us consider again the transactions of example 1. According to the above definition, and by exploiting the similarity values computed in example 3, we can evaluate similarity between transactions wps_{s_2} and wps_{s_8}. To this purpose, we also assume that the corresponding viewing time sequences are $vts_{s_2} = \langle 8, 3, 10 \rangle$ and $vts_{s_8} = \langle 5, 6, 7 \rangle$. Finally, we obtain:

$$sim(s_2, s_8) = 0.6453$$

Notice that traditional similarity measures based on the vector-space model fail in capturing the dissimilarities between such sequences. For example, the Jaccard similarity (adopted, e.g., in [8, 9]), computes distance 0. □

5 Exploiting Profiles

The process of personalized recommendation delivery is based on the extraction of browsing patterns from user transactions. Precisely, first transaction clusters are found, then user profiles are associated to these clusters. Transaction clusters are subsets of users with similar browsing behavior. Each cluster is a source of recommendations for all those current users whose behavior can be considered similar to the transactions within the cluster itself: recommendations for a given user are detected based on the pages accessed by other users with similar navigational behavior. User profiles are cluster representatives, which are exploited to synthetically describe the typical browsing behavior of all users within the corresponding transaction clusters. They are leveraged to efficiently verify whether or not current user activities can be classified into a given cluster.

5.1 Knowledge Extraction from Usage Data: Profile Learning

Clusters of similar user transaction are formed by means of an approach introduced in [1]. Similarity values are computed for all pairs of user transactions. These values are then used to form a user transaction graph $TG = \langle V, E \rangle$, where nodes in V correspond to transactions and arcs in E represent similarity degrees among transactions. Arcs are in fact weighted by the similarity values associated to their end nodes. User transactions are then clustered using a graph partitioning approach based on the algorithm Metis [13], which efficiently partitions very large data sets.

$\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ is the set of all the detected clusters. Such clusters contain similar transactions and are, therefore, indicative of common browsing patterns of as many groups of users. Moreover, they are extremely useful in order to derive user profiles which synthetically describe the site usage, the navigation goals and the content preferences of the above subsets of users.

In order for a Web site to support user navigation, by proactively driving them to the Web pages that meet their browsing requirements, it is necessary to dynamically classify current user

sessions into at least one of the existing clusters. Recommendations regarding pages not yet visited by current users are then extracted from the profiles associated to these clusters.

In our approach, we assume that cluster profiles are sequences. This approach to user profiling substantially differs in terms of profile structure with respect to most of traditional approaches [20, 21], where each profile is modelled as n dimensional vectors defined over the space of the Web pages within a particular Web site. Each dimension of such vectors corresponds to a specific Web page and contains the weighted relevance of that page within the profile itself. However, since these representations cannot describe the sequential nature of user browsing activities, vector models fail to accurately describe the typical navigational behavior of subsets of users. The main benefit of modelling profiles as sequences is that these representations take into consideration the actual trajectories followed by users while traversing a Web site. Browsing trajectories contain essential, implicit meta-information about user navigational behavior, such as the contexts in which Web pages are accessed. Page context information effectively allow to detect substantial differences, in terms of either information requirements, preferences or browsing goals, among seemingly similar navigational behaviors of users accessing the same pages. This contributes to enhancing the effectiveness of the personalization process, since it allows to tailor the delivered recommendations to meet user-specific information requirements.

Example 5. Users following trajectories wps_{s_2} and wps_{s_8} in example 1 exhibit different behaviors, though accessing the same pages. For instance, users following trajectory wps_{s_2} could simply need to read some information conveyed by page p_2, thus using both p_1 and p_3 as navigational pages. On the contrary, users following trajectory wps_{s_8} could need information which are distributed on pages p_2 and p_1 and only use p_3 as a navigational page.

User profiles are built based on an approach conceived to detect and suitably modify cluster medoids. Given a usage cluster $\mathcal{C}_i$, a transaction $m_{\mathcal{C}_i} \in \mathcal{C}_i$ is found such that the quantity

$$ics_{\mathcal{C}_i}(m_{\mathcal{C}_i}) = \sum_{t' \in \mathcal{C}_i} sim(t', m_{\mathcal{C}_i})$$

(representing the intra-cluster similarity value associated to $\mathcal{C}_i$) is maximized. However, since medoids coincide with specific cluster elements, they tend to be too detailed cluster representatives, thus not effectively representing all other transactions within the same clusters. Hence the need for *generalized medoids*. In order to generalize a medoid $m_{\mathcal{C}_i}$, thus making it an effective cluster representative, two different steps are performed:

- the removal of all pages in $wps_{m_{\mathcal{C}_i}}$ not frequently accessed within the user sessions in $\mathcal{C}_i$;
- the replacement of the viewing times in $vts_{m_{\mathcal{C}_i}}$ with the average viewing times, computed by taking into account all user sessions within $\mathcal{C}_i$.

To these purposes, two kinds of n dimensional vectors, $v_{\mathcal{C}_i}$ and $t_{\mathcal{C}_i}$, are associated to each cluster $\mathcal{C}_i \in \mathcal{C}$. Each dimension of such vectors correspond to a specific Web page: $v_{\mathcal{C}_i}$ is a vector of page modes, while $t_{\mathcal{C}_i}$ contains the average viewing times for the Web pages in $\mathcal{U}$. All Web pages not frequently accessed by users in cluster $\mathcal{C}_i$ are detected within medoid $m_{\mathcal{C}_i}$ by leveraging $v_{\mathcal{C}_i}$ and a user-defined frequency threshold τ. Precisely, all elements within the mode vector whose access frequency is less than τ correspond to pages to be removed from $m_{\mathcal{C}_i}$. Finally, the viewing time of each page in $wps_{m_{\mathcal{C}_i}}$ is replaced, within $vts_{m_{\mathcal{C}_i}}$, with the average viewing time of the page itself. The set of all cluster profiles is formally defined as $\mathcal{P} = \{m_{\mathcal{C}_1}, \ldots, m_{\mathcal{C}_k} | \mathcal{C}_1, \ldots, \mathcal{C}_k \in \mathcal{C}\}$. Details of the algorithm for the generalization of cluster medoids are given in fig. 3.

Input: A set $\mathcal{S} = \{s_1, \ldots, s_N\}$ of sessions; a threshold τ.

Output: a *generalized medoid* m.

Method: Perform the following steps:

1. let $m = \min_{t \in \mathcal{S}} ics_{\mathcal{S}}(t)$;
2. Compute the vector $v_{\mathcal{S}}$ of access frequencies of the web pages contained in the sessions within $\mathcal{S}$.
3. Compute the vector $t_{\mathcal{S}}$ containing the average viewing times of all pages within the sessions in $\mathcal{S}$.
4. Let $E = \{p_{i_1}, \ldots, p_{i_r}\}$ be the set of all unfrequent pages (according to τ), sorted by ascending frequency. Repeat until no more changes are detected within m:
   - if $p_j \in wps_m$ such that $p_j \in E$ then let m' be a sequence which contains every page access in m but all occurrences of p_j (and its associated viewing times)
   - if $ics((m') \geq ics_{\mathcal{S}}(m)$
     - then $m := m'$
5. for each viewing time $t_i \in vts_m$ associated to a corresponding Web page $p_i \in wps_m$ do:
   - replace t_i with $t_{\mathcal{S}}[p_i]$ (that is, with the average viewing time of p_i in $\mathcal{S}$)
6. return m

Fig. 3. Computing a Generalized Medoid

5.2 Delivery of Personalized Recommendations

User sessions can be conceptually divided into a number of sub paths, each characterized by a different navigation goal. The process of delivering personalized recommendation is based on current users' most recent navigation history: that is, recommendations are thought to help users achieve their current sub path requirements. Current users' recent navigation history maps to the last n visited Web pages (usual values of n range from 2 to 4, as it is pointed out in [20, 21]).

The process of delivering personalized recommendations is detailed next. As formalized in sec. 2, assume that a current user u requests a Web page $p \in \mathcal{U}$. As soon as the request for p is detected by the Web site, rh, that is the most recent browsing history, of u, is compared with the cluster profiles in order to detect what pages are potentially of interest to her/him.

The main reason for considering all the cluster profiles is that, potentially, a user may exhibit a navigational behavior which is not entirely reflected by any profile in $\mathcal{P}$, but at the same it may have some similarities in common, though at different degrees, with a number of the above profiles. The subset of all cluster profiles which are found to best reflect the navigation activities of u can be expressed as

$$ClosestProfiles = \{m_{\mathcal{C}_i} \in \mathcal{P} | sim(rh, m_{\mathcal{C}_i}) \geq \vartheta\}$$

Parameter ϑ is a user defined threshold, which is leveraged to act on *ClosestProfile* size. It is a trade off between system recommendation effectiveness and delivery efficiency.

Web pages within each cluster profile $m_{\mathcal{C}_i} \in ClosestProfiles$ are assigned a value representing the expected interest of u in such pages. In general, interest in a page p depends on the importance of that page within its cluster profile $m_{\mathcal{C}_i}$, the similarity degree between user most recent navigation history rh and m_{C_i} itself, and the physical link distance between p and any page in rh. This is quantified by the formula below, which defines the *page interest indicator* of a page p not yet visited:

$$pii(p, m_{\mathcal{C}_i)} = significance(p, m_{\mathcal{C}_i}) \times sim(rh, m_{\mathcal{C}_i}) \times (\log(distance(p, rh)) + 1)$$

If p has been already accessed anywhere within the current user session then $pii(p, m_{\mathcal{C}_i}) = 0$.

The distance factor is computed as the minimum path length between p and any page in rh in the graph representing the site structure[1]. Factor $significance(p, m_i)$, in turn, represents the significance weight of p within $m_{\mathcal{C}_i}$. In general, significance of page p is assumed to depend both on its occurrence frequency within a given profile sequence $s = \{(p_1, t_1), \ldots, (p_m, t_m)\}$ and on key information such as where and for how long it has been accessed within s.

$$significance(p, s) = \frac{occurrences(p, s)}{len(wps_s)} \times \frac{\sum_{(p_i,t_i) \in s | p_i = p} cw(i) \frac{t_i}{length(p_i)}}{\sum_{(p_j,t_j) \in s} cw(j) \frac{t_j}{length(p_j)}}$$

In the above definition, $length(pg)$ indicates the size (in bytes) of a page pg. Page position information is leveraged by a context weight function $cw(n)$ to devise a suitable weighting for page contexts within navigation sequences: according to the particular application domain, page significance could also depend on an *ad hoc* page position weighting. For instance, in a typical information

[1] A site structure graph is a directed graph in which nodes are represented by pages, and arcs by links between pages (as they appear within the page contents).

search scenario, starting or ending sequence pages could be considered as more significative than those in the middle of the sequence itself, as they could respectively represent the first attempt to find useful information and the final step usually characterized by a result enjoying phase.

The amount of time spent visiting p within s with respect to the total time duration of s itself is an essential element for the detection of user interest. In general, in fact, the longer a user visits a page, the likelier that user is interested in that page. However, there are cases in which this intuition may lead to unreliable outcomes. A short viewing time does not necessarily mean that the corresponding page is of no interest for a certain user: simply that page may have a short length, thus requiring reduced viewing times. For this purpose, viewing times are normalized by page length.

Finally, a list of recommendations is generated by choosing all the pages within the profiles in *ClosestProfiles*, which have *pii* values over a user defined threshold μ[2]:

$$RecList = \{p \in m_{\mathcal{C}_h} | \mathcal{C}_h \in ClosestProfiles \text{ and } pii(p, m_{\mathcal{C}_i}) \geq \mu\}$$

Obviously, if a page in *RecList* is contributed by various *ClosestProfiles* profiles, it appears in that list with its maximum *pii* value. Personalized recommendations in *RecList* are sorted by the expected interest of current user u in the selected pages and are then added with the same order to the page p requested by u.

6 Related Works

Traditional approaches to personalized recommendation systems rely on three main classes of technologies:*i* collaborative filtering, *ii* content filtering and *iii* Web usage mining.

Collaborative filtering technology [14, 19, 28, 11] generates recommendations for a target user by first detecting a neighborhood of k closest users who have rated Web items in a manner similar to that of the target user. Hence, the neighborhood as a whole is exploited to recommend items not yet visited by the target user. Some typical limitation negatively affecting collaborative filtering technology are pointed out in [23, 27]. In our opinion, two main issues make the approach uneffective.

- The collaborative filtering approach is based on explicit user input which results in subjective data and static user profiles. Subjective data pose the problem of user profile reliability, while static user profiles cause poor personalization accuracy as such profiles age.
- As neighborhood formation and recommendation delivery are both on-line activities, collaborative filtering based recommendation systems suffer from limitations in terms of efficiency and scalability when the number of both users and items to recommend increases.

[2] A method for outlier detection can be exploited here.

Systems based on content-filtering [15, 22, 12] learn a model of user interests and then try to estimate actual user interest in individual documents based on similarity between these documents and the learned profiles. For example, Letizia [15] is a client-side agent which operates on top of any Web browser. It first learns a model of user interests by tracking a user's browsing behavior: documents which are bookmarked, repeatedly visited or read for a sufficiently long time (with respect to their length) are used to infer information on user interests. The agent then autonomously and in parallel with user browsing searches for interesting documents close to current user position. Recommendations are continuously generated in order to suggest the user what Web pages should best meet her/his need. A major limitation of content-filtering systems is that they do not deal with usage data, which are extremely useful to discover correlations both among users (from a behavioral point of view) and site pages/items (from a usage point of view).

Recent research efforts [16, 20, 21, 18, 3, 17] have focused on Web usage mining techniques in order to automatically discover common behavioral patterns from usage data and learn profiles for groups of users. In [3] an approach to the design of adaptive Web sites based on Web usage mining techniques is presented. Recommendations for user navigation support are extracted from page clusters, which capture pages related through usage based on their co-occurrence patterns across user transactions. Such clusters are obtained by first discovering all interesting frequent page sets (i.e. group of pages frequently occurring together in many user transactions) which exist in the transaction dataset. Then a hypergraph is formed, where nodes correspond to Web pages and hyper arcs to the above frequent page sets. Finally, the hypergraph is partitioned by means of a technique called Association Rule Hypergraph Partitioning (ARHP). Two differences of [3] with respect to the proposed formalization are discussed next.

- The recommendation delivery process is conceived to extract recommendations from a given cluster based on usage similarity between the pages recently visited by a certain current user and those within the corresponding cluster profile: content similarity between such pages is not taken into consideration while evaluating how each cluster matches current users' recent navigation history. This negatively affects the accuracy of the delivery process.
- User sessions are modelled by means of binary vectors, which do not accurately reflect current users' actual interests, since all pages visited by a certain user appear as equally interesting to that user.

[20] provides a comparison between two techniques for deriving *aggregate* usage profiles: Profile Aggregations based on Clustering Transactions (PACT) and ARHP. PACT works on transaction clusters: each such cluster represents a subset of users with similar navigation behavior. Precisely, PACT is based on clustering similar user transactions. ARHP (described above) generates

page clusters which, on the contrary, capture overlapping interests of different kinds of users. An aggregate cluster profile is a representative of all transactions within a given cluster. These representatives are n-dimensional vectors over a space of Web pages: the i-th dimension represents the corresponding Web page within that site. In the case of PACT, profiles are obtained computing for each transaction cluster its mean vector, while with ARHP each dimension of a profile vector for a given cluster is the connectivity value of the corresponding Web page within that cluster. The main difference between the two techniques is that profiles obtained through PACT represent pages that frequently occur together across similar user transactions, while ARHP results in cluster whose profiles highlight pages accessed together across user transactions, even if such transactions are not similar. The main differences between PACT [20] and our proposal are pointed out next.

- User sessions both active and already terminated are modelled as vectors. This means that there is a lack of information about the chronological order with which past users have visited the pages within their corresponding sessions.
- Profiles for groups of users are computed as mean vectors: that is, cluster representatives are conceived as transactions not corresponding to any actual navigation path within the cluster. Moreover, while transaction profile ought to highlight the typical navigation behavior which is common to all user within the cluster associated to that profile, vector representations achieve this goal only partially. In fact, typical user navigation behavior is synthesized to be only a set of visited pages, which can be not necessarily within the same session. As a consequence, both information on the chronological order of page accesses and locality is lost.
- Current user browsing history and cluster representatives are compared by leveraging a normalized cosine similarity measure which compares the two kinds of vectors only on the basis of their dimension values, i.e. the pages contained. Our approach leverages a notion of sequence similarity which is parametric to the sequence topology.

[21] presents an interesting approach conceived to improve the effectiveness of the personalization process. The idea is that users should receive recommendations about pages which exhibit either usage or content similarities with those within their recent navigation history. To this purpose, the concept of content profiles is introduced: such profiles are representatives of (possibly overlapping) clusters grouping Web pages with partly related contents. Recommendations are extracted from both transaction and content profiles, which are defined as n-dimensional vectors over the space of a site Web pages. Besides those already discussed with respect to [20], another main difference between [21] and our approach is discussed next.

- The matchings of current users' recent navigation history with usage and content profiles are evaluated independently. In our approach, on the contrary, the process of matching user

recent clickstream with transaction profiles is based on an approach conceived to detect page synonymies from both a content and usage point of view. Pages respectively within user recent browsing history and transaction profiles are considered to be similar if they have similar contents and/or are used in a similar fashion.

The notion of Adaptive Web Site (AWS) in introduced in [24]. AWSs are defined as Web sites that are capable of learning user expectations from their access patterns. Based on such information AWSs can automatically improve both their internal organization, in order to facilitate user navigation, and data presentation. Two kinds of approaches are discussed only from a conceptual point of view :

- customization, i.e., the tailoring of site interface presentation to a specific user.
- optimization, i.e., an attempt to make the usage of Web sites easier for all kind of users (customization on the contrary focuses on individuals), even occasional visitors.

[25] shows how knowledge extracted from usage data can be fruitfully exploited to make a Web site usage easier in response to user actions. To this purpose a number of transformations are presented.

Two specific methods for clustering sequential data are shown in [2, 10]. In [2] a probabilistic framework for clustering sequences based on a variant of EM (expectation-Maximization) algorithm is presented. The algorithm is applied to learn a mixture of first-order Markov models. The approach is attractive and scales linearly with the number of clusters and with the number of available, but in our opinion a drawback of the approach is the lack of i a notion of cluster rapresentative, and ii a matching criterion between a current user session and the cluster partition containing the most suitable profile.

The approach developed in [10] avoids explicitly defining a suitable notion of similarity between sequences, and instead is based on the extraction and comparison of significant features (e.g., sequential patterns [29]) from the available sequences. The approach is based on a critical evaluation of time-warping distance, based on efficiency considerations. Although in our approach we adopt the time-warping distance, our analysis is based mainly on effectiveness considerations, which can be strongly influenced by even a weak similarity between sequences of pages not sharing any sequential pattern: according to [10], in fact, such sequences have no features in common, and hence, contrarely to our approach, they exhibit a high dissimilarity. Notwithstanding, we plan a more detailed experimental comparison.

7 Conclusions and future works

A conceptual methodology for the delivery of personalized recommendations has been proposed which leverages a number of Web usage mining techniques to automatically learn profiles for groups of users, without requiring explicit user collaboration and, consequently, avoiding any evaluation of subjective data. Moreover profiles can be automatically updated, which avoids poor recommendation delivery performances as profiles age. The main contributions are summarized below.

1. A new approach to the discovery of transaction cluster profiles, based on *generalized medoids*, which makes it possible to describe the general browsing behavior for a group of users as a sequential pattern that frequently appears in the cluster under investigation.
2. The definition of a new methodology for computing similarity between Web pages based on the detection of both content and usage synonymies, which let any data mining engine to discovery actual degrees of similarity even among pairs of user sessions having no pages in common.
3. The introduction of a novel approach to computing similarity between user sessions, which leverages Web page contexts to take into consideration the topology of user navigation sequences.

A number of experiments are being performed in order to quantify both the reliability of generalized medoids as representatives of transaction clusters, and the accuracy of the function for evaluating similarity between user sessions. Also, the proposed approach to personalization is being compared, in terms of clustering, profiling and recommendation effectiveness and efficiency, with conventional approaches based on vector representations of both user sessions and cluster profiles.

References

1. A. Banerjee and J. Ghosh. Clickstream Clustering using Weighted Longest Common Subsequences. In *Proc. of the Web Mining Workshop at the 1st SIAM Conference on Data Mining, Chicago*, 2001.
2. I.V. Cadez, S. Gaffney, and P. Smyth. A General Probabilistic Framework for Clustering Individuals and Objects. In *Proceedings of the ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, USA, August 2000.
3. B. Mobasher R. Cooley and J. Srivastava. Creating Adaptive Web Sites Through Usage-Based Clustering of URLs. In *Proc. IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, 1999.
4. R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.
5. R. Cooley, B. Mobasher, and J. Srivastava. Grouping web page references into transactions for mining World Wide Web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1):5–32, 1999.

6. Robert Cooley. *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. PhD thesis, University of Minnesota, 2000.
7. C. Fox. *Lexical analysis and stoplists*. Prentice Hall, 1992.
8. F. Giannotti, C. Gozzi, and G. Manco. Clustering Transactional Data. In *Proc. 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 175–187, 2002.
9. S. Guha, R. Rastogi, and K. Shim. ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems*, 25(5):345–366, 2002.
10. Valerie Guralnik and George Karypis. A Scalable Algorithm for Clustering Sequential Data. In *Proc. IEEE International Conference on Data Mining*, pages 179–186, 2001.
11. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proc. of the 1999 Conference on Research and Development in Information Retrieval*, 1999.
12. T. Joachims, D. Freitag, and T.M. Mitchell. Web Watcher: A Tour Guide for the World Wide Web. In *Proc. 15th International Joint Conference on Artificial Intelligence, (IJCAI97)*, pages 770–777, 1997.
13. G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.
14. J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. GroupLens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
15. H. Lieberman. Letizia: An Agent that Assists Web Browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 924–929, 1995.
16. B. Mobasher, R. Cooley, and J. Srivastava. Automatic Personalization Through Web Usage Mining. Technical Report TR99-010, Computer Science Department, De Paul University, 1999.
17. B. Mobasher, R. Cooley, and J. Srivastava. Automatic Personalization Based On Web Usage Mining. *Communications of the ACM*, 43(8):142–151, 200.
18. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective Personalization Based on Association Rule Discovery from Web Usage Data. In *Proc. 3rd ACM Workshop on Web Information and Data Management (WIDM01)*, 2001.
19. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. Technical Report 00-005, Computer Science Department, De Paul University, 2001.
20. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. *Data Mining and Knowledge Discovery*, 6(1):61–82, January 2002.
21. B. Mobasher, H. Dai, T. Luo, Y. Sung, and J. Zhu. Integrating Web Usage and Content Mining for More Effective Personalization. In *Proc. 1st International Conference on Electronic Commerce (ECWeb00)*, pages 165–176, 2000.
22. D.S.W. Ngu and X. Wu. Sitehelper: A Localized Agent that Helps Incremental Exploration of the World Wide Web. In *Proc. 6th International World Wide Web Conference*, 1997.
23. M. O'Conner and J. Herlocker. Clustering items for collaborative filtering. In *Proc. of the ACM SIGIR Workshop on Recommender Systems, Berkley, CA*, 1999.

24. M. Perkowitz and O. Etzioni. Adaptive Web Sites: An AI challenge. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI97)*, pages 16–23, 1997.
25. M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically learning from user access patterns. In *Proc. of the Sixth International WWW Conference*, 1997. Available at `http://www.scope.gmd.de/info/www6/posters/722/index.html`.
26. G. Salton and M.J. McGill. *Introduction to Modern Information Retrivial*. McGraw-Hill, 1983.
27. B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis or Recommender Algorithms for E-Commerce. In *Proc. of the 2nd ACM E-Commerce Conference (EC'00), Minneapolis*, 2000.
28. U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proc. of CHI 95 Conference*, pages 210–217, 1995.
29. R. Srikant and R. Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. In *Proc. Int. Conf. on Extending Database Technology (EDBT96)*, volume 1057 of *Lecture Notes in Computer Science*, pages 3–17, 1996.