Learning Web Users Profiles With Relational Clustering Algorithms

Nicolas Labroche
Université Pierre et Marie Curie, Paris 6
Laboratoire d'Informatique de Paris 6, UMR CNRS 7606
4, Place Jussieu, Case 169, 75252 Paris cedex 05, France
nicolas.labroche@lip6.fr

Abstract

In the context of web personalization and dynamic content recommendation, it is crucial to learn typical user profiles. Although there exists several approaches to mine user profiles (such as association rules or sequential patterns extraction), this paper focuses on the application of relational clustering algorithms on web usage data to characterize user access profiles. These methods rely on the definition of a distance (or dissimilarity) measure between user sessions and thus can carry more information (content, sequence of page views, context of navigation) than simple transactions. Moreover, as web user sessions are often noisy, uncertain or inaccurate (because of proxy web server, local browser cache and sessions building heuristics), we propose to use two clustering algorithms: the leader Ant clustering algorithm that is inspired by the chemical recognition system of ants and a new variant of the fuzzy C Medoids. The paper also describes the similarity measures used to compare these algorithms with the traditional fuzzy C Medoids on real web usage data sets from French museums. The evaluation is conducted according to the quality of the output partitions and the interpretability of each cluster based on its content.

Introduction

In the context of web personalization and dynamic content recommendation, it is crucial to learn typical user profiles. These profiles serve as user models in many recommender systems based on collaborative filtering technique. Thus, the method heavily relies on the ability of the clustering algorithm to mine groups of homogeneous navigations that correspond to the expression of well defined web users information needs. Although there exists several approaches to mine user profiles - such as association rules or sequential patterns extraction -, this paper focuses on the application of relational clustering algorithms on web usage data to characterize user access profiles. These methods rely on the definition of a distance (or dissimilarity) measure between user sessions and thus can carry more information (content, sequence of page view, context of navigation) than simple transactions. Moreover, as web user sessions are often noisy, uncertain or inaccurate (because of proxy web server, local browser cache and sessions building heuristics), we propose to use two clustering algorithms: the leader Ant clustering algorithm that is inspired by the chemical recognition system of ants and a new variant of the fuzzy C Medoids.

The application reported in this paper aims at extracting and analyzing web user navigation patterns on web sites of French institutions related to sciences and technology (like museums, aquariums ...). The objective of the study was not to evaluate the ergonomy or the functionalities of the web sites, but to point out the differences that may exist between the role of the physical institution and its virtual counterpart, and to eventually enlighten behaviors observed on all the web sites that may be typical of this activity.

This paper is organized as follows: the second section motivates our work by recalling some of the previous researches that have been conducted in the web Usage Mining field. The third section describes the leader Ant and our new variant of fuzzy C Medoids. The forth section presents the similarity between web user sessions and describes the results of the experiments that have been conducted to evaluate our algorithms. Finally, section five discusses the results and briefly develops some of the perspectives of this research work.

Related Work on Web Usage Mining

The analysis of web user behaviors is known as web Usage Mining (WUM) that is to say, the application of data mining techniques to the problem of learning web usage patterns. The WUM is a relatively new research field that aims at understanding the navigation of users on web sites by inferring their objectives and their motivations from the stream of requests made during their navigation sessions or any other interaction with web sites (inserting or editing text in a web page (Kay *et al.* 2006), printing documents ...).

Many works have been conducted in the last ten years to extract, analyze, model or predict the web users information needs on a web site. Cooley et al. (Cooley, Srivastava, & Mobasher 1997) proposed some heuristics to prepare the

web log file, to filter and to reconstruct the web sessions. Then, numerous approaches have been proposed to help understanding the web users behaviors and inferring their motivations from their requests on web servers. Cooley et al. (Cooley, Mobasher, & Srivastava 1999) introduced the webMiner system that allows to filter web log files, to reconstruct web sessions as transaction vectors, to compute association rules from the sessions and to request the set of rules with an SQL-like language so that an expert of the web site can extract easily meaningful information. Similarly, Spiliopoulou et al. (Spiliopoulou & Faulstich 1998) described the webWUM system, which also introduces an SQL-like language to request new behavioral rules or mine from an aggregated tree representation of the web sessions. This work has been extended in (Spiliopoulou, Pohle, & Faulstich 1999) where the authors distinguish profiles from episodic or first time visitors, usual visitors and clients to apply modifications on the web site to increase the number of visitors that are clients. Masseglia et al. (Masseglia, Poncelet, & Cicchetti 1999; Masseglia, Poncelet, & Teisseire 1999) proposed webTool, an expert system that relies on sequential patterns extraction and uses the incremental method ISEWUM. The system aims at reorganizing web sites or at predicting web pages like Davison (Davison 2002; 1999). Perkowitz and Etzioni (Perkowitz & Etzioni 1999) reorganize web sites via the generation of a thematic index page using a conceptual clustering algorithm named PageGather.

Some other works (Labroche, Monmarché, & Venturini 2003; Baraglia & Palmerini 2002; Heer & Chi 2001; Fu, Sandhu, & Shih 1999) apply clustering algorithms to discover homogeneous groups of (web) sessions. The underlying idea of these methods is that the algorithm should group in the same cluster the sessions that correspond to the users that navigate similarly, that is to say, that have the same motivation and interests. The objective can be either to discover users that accessed the same resources on the web site or to find clusters of web pages that co-occur frequently in the same sessions. This kind of approach allows defining a profile of typical web site access from each discovered cluster and can highlight information that is not directly accessible in the data. The extracted profiles can be used for web pages recommendation purposes or dynamic web site organization.

Yan et al. (Yan *et al.* 1996) use the First Leader clustering algorithm to create groups of sessions. The sessions are described as hits vectors in which each component corresponds to a web page and indicates the number of times it has been accessed during the session. The weakness of the method stands in the use of the First Leader algorithm. Although very fast, it is dependant on the order in which web sessions are processed and may need to parameter its maximal number of clusters. Estivill-Castro et al. (Estivill-Castro & Yang 2001) uses a k-Means-like algorithm that relies on a median rather than a mean to estimate the cluster centers. In (Nasraoui *et al.* 2002), the authors use a fuzzy C Medoids approach, derived from the fuzzy C Medoids algorithm FCMdd described in (Krishnapuram *et al.* 2001), to deal with the uncertainty and inaccuracy of the web sessions. The new algorithm is a linear version of FCMdd algorithm that takes into account only the p objects that most belong to a cluster to compute its new medoid.

Nevertheless, a problem still remains: the analyst has to specify the number of expected clusters (even if it is possible to overestimate this number and only keep the most relevant clusters at the end). To automatically evaluate this number of expected clusters Heer and Chi (Heer & Chi 2002) propose a simple heuristic that computes the stability of the partitions for different number of clusters. The method works well but is extremely time consuming and thus may not be applicable in a real study context. In (Nasraoui *et al.* 1999), the authors propose the CARD relational clustering algorithm (Competive Agglomeration for Relational Data) that is able to determine the appropriate number of clusters starting from a large number of small clusters. In (Suryavanshi, Shiri, & Mudur 2005), the authors propose an incremental variant of a subtractive algorithm that allow to update partitions from previous analysis. This method determines automatically the number of clusters according to an estimation of the potential of each object to be a cluster center based on the density of the other objects in its neighborhood.

Labroche et al. (Labroche, Monmarché, & Venturini 2003) describe a relational clustering algorithm inspired by the chemical recognition system of ants named AntClust. In this model, each artificial ant possesses an odor representative of its nest membership called "label" and a genome, which is associated with a unique object of the data set. The algorithm simulates meetings between artificial ants according to behavioral rules to allow each ant to find the label (or nest) that best fits its genome. AntClust does not need to be initialized with the expected number of clusters and runs in linear time with the number of objects. AntClust has also been successfully applied to the web sessions clustering problem with various representations of sessions. More recently, Labroche (Labroche 2006a; 2006b) introduces the Leader Ant algorithm that relies on the same biological model and that is between 5 to 8 times faster than AntClust with similar clustering error values on benchmark data sets.

Recently, Mobasher (Mobasher 2006) proposed an overview of the main data mining approaches that have been used to mine user profiles in a web personalization perspective (association rules, sequential patterns and clustering methods).

The Leader Ant and Robust Fuzzy C Medoids Algorithms

This section briefly describes the two relational clustering algorithms that we use for our analysis: the Leader Ant clustering algorithm (LA) and the new variant of Fuzzy C Medoids.

The Leader Ant Algorithm

LA aims at solving the unsupervised clustering problem by reproducing the main principles of the chemical recognition system of ants. Nevertheless, the underlying model of LA, although inspired by real ants system, has been adapted to match more specifically the objectives of the clustering problem and for performance purposes. In LA, an artificial ant is described by three parameters:

1. The genome is associated with an unique object of the data set;
2. The template is the same for all artificial ants and is defined as the mean similarity observed between randomly chosen ants;
3. The label reflects the nest membership of each artificial ant.

LA is a one-pass relational agglomerative algorithm that iteratively selects at random a new ant a (that has not been already assigned to a nest), and determines its label or nest membership by simulating meetings with randomly selected ants from each existing nest. During these meetings, the ant a estimates the similarity of its genome with those of ants from the evaluated nest. At the end, the ant a joins the nest with most similar ants or build its own new nest if the observed similarity values are under its template threshold.

Finally, when all ants are assigned to a nest, the smallest nests can optionally be deleted and their ants reassigned to the most similar clusters.

The idea of the LA algorithm is to replace the centroid or medoid computation of k-Means like approaches by random meetings between ants to decide if the object associated to the ant should belong to the cluster or not. The LA algorithm runs in linear time with the number of objects n and the number of clusters k, i.e. in $O(kn)$ as each object is compared to each existing cluster to decide its assignment.

LEADER ANT ALGORITHM
Input: A data set with n objects
Output: A partition of the n objects

(1) Initialization of Artificial Ants
(2) Assignment of one object to each ant
(3) Template computation
(4) Iterative Building of Nests
(5) Selection of an artificial ant
(6) Random meetings with ants from each nest
(7) Estimation of similarity with each nest
(8) Assignment to a nest or building a new nest
(9) Deletion of Smallest Nests (optional)

A New Variant of Fuzzy C Medoids Algorithm

As our algorithm is a variant of the Fuzzy C Medoids (FCMdd) (Nasraoui *et al.* 2002; Krishnapuram, Joshi, & Yi 1999), this section first recalls the main principles of the FCMdd algorithm before introducing the specificity of our new approach.

The FCMdd algorithm is a variant of the fuzzy C Means algorithm introduced by Bezdek (Bezdek 1981) that replaces the original mean computation by a medoid computation because it is a more robust estimator to noise and outliers. As stated by (Krishnapuram, Joshi, & Yi 1999), "the objective functions are based on selecting C representative objects (medoids) from the data set in such a way that the total dissimilarity within each cluster is minimized".

Initially, the algorithm is set with C randomly chosen medoids (several heuristics have been proposed to improve this initialization). Then the algorithm iteratively computes a membership matrix and defines new medoids until the algorithm converges or a stopping criterion is met. The main limit of this approach is that its complexity is quadratic with the number of objects due to the computation of the medoid for each cluster. In (Nasraoui *et al.* 2002; Krishnapuram *et al.* 2001), the authors propose three medoids initialization heuristics and one improvement named LFCMdd (Linear FCMdd) that considers only the p objects ($p \ll$ number of objects) that have the highest membership value in a cluster to compute the medoid. The complexity is reduced to $O(npc)$ with n, the number of objects and c, the number of clusters. Thus, the problem resides in the choice of an adapted value for p: a large value loses the benefit of the approach but enables a better search for the medoids, and a small value can lead to quicker convergence but poorer medoid definition.

LINEARIZED FUZZY C-MEDOIDS ALGORITHM (LFCMDD)
Input: A data set X with n objects
Output: A partition of the n objects

(1) Fix the number of clusters c; $iter = 0$
(2) Pick the initial set of medoids $V = \{v_1, v_2, \ldots, v_c\}$
(3) **repeat**
(4) Compute memberships $u_{i,j}, \forall i \in [1, c]$ and $\forall j \in [1, n]$
(5) Identify $X_{(p)i}$ the set of p objects with highest membership for cluster $i \in [1, c]$
(6) Store the current medoids $V^{old} = V$
(7) Compute the new medoids $v_i, \forall i \in [1, c]$
(8) $q = argmin_{x_k \in X_{(p)i}} \sum_{j=1}^{n} u_{i,j}^m \times r(x_k, x_j), v_i = x_q$
(9) iter = iter + 1
(10) **until** $V^{old} = V \vee iter = MAX\_ITER$
(11)

The limitations of this method are that it introduces a new parameter p, that need to be set according to the input data set, and also that for each cluster the algorithm need to sort the membership values of the n objects to keep only the p highest values, which induces the use of an efficient

sorting algorithm not to lose the benefit in complexity of this approach.

Our algorithm uses an adaptation of the heuristic by Estivill-Castro et al. (Estivill-Castro & Yang 2001) that was initially designed to develop a robust k Medoids algorithm. Our idea is to adapt this method to the fuzzy C Medoids algorithm. This method allows to find the representative object of a cluster with $n \times \sqrt{n}$ computations for a data set of size n.

The basic idea is to separate the whole data set into $\sqrt{n}$ sub-data set of approximately $\sqrt{n}$ objects each, and then to apply the exhaustive medoids search method on each sub-data set. At the end, the method produces $\sqrt{n}$ candidate medoids and their associated scores for each cluster.

The score of a candidate, in the context of a fuzzy clustering algorithm, is the sum of the distances between the candidate and all the other objects weighted by the membership of the objects in the considered cluster. The following equation formalizes the method to compute the score $s_k(c)$ of each candidate c for a cluster k:

$$s_k(c) = \sum_{\forall o \neq c \in O} u(o,k)^m \times r(o,c) \qquad (1)$$

where O is the set of all objects ($|O| = n$), r is a distance measure between the object o and the candidate medoid c for the cluster k, u is the membership matrix and m is a fuzzy factor whose value is greater than 1 and that gives more or less importance to the membership value.

The candidate with the smallest score value, i.e. that minimizes the distances with the other objects, is defined as the new medoid for the cluster. It is important to note that in our implementation of the algorithms, it is allowed for one cluster to choose an other cluster's medoid as medoid, which may lead to degenerated partitions.

The convergence of our algorithm, like the other iterative clustering algorithms (k-Means, k-Medoids ...), heavily depends on the quality of the initial partition of the data. We use the same initialization scheme as Nasraoui et al. (Nasraoui *et al.* 2002) in which a first medoid is computed as the most central object among the set of objects and then the $c-1$ other medoids are defined as the objects that have the maximal minimum distance with existing medoids. Our approach runs in sub-quadratic time complexity ($O(cn^{\frac{3}{2}})$), needs less parameter than LFCMdd, and has proved to be efficient on artificial and real numerical benchmark data sets. The next section presents the comparative results of our new algorithm that we named hereafter VFCM (Variant of Fuzzy C Medoids) with LFCMdd and the leader Ant on real web usage data.

Experiments and Results

This section presents the results of different studies that have been conducted on real web sessions from three museums web sites, namely: the museum of Bourges (Museum of Bourges 2007), Cap Sciences (Cap Sciences 2007) and CCSTI of Grenoble (CCSTI Grenoble 2007).

The section first describes the preprocessing we applied on web log files to build the corresponding web sessions. Second, it introduces our representations of web sessions either as sequences or as a set of unordered web pages. Then, this section presents several experimental studies that aim at evaluating the quality of our approach. The first study describes an experiment that aimed at evaluating the convergence of our algorithms (leader Ant and VFCM) on real web data sets to evaluate their applicability. The second study points out the differences between the partitions obtained with both similarity measures on the Cap Sciences web site.

Finally, this section reports the results obtained by the previous algorithms and compares their efficiency in building meaningful user profiles.

Preprocessing of Web Log Files

The web log files have been converted from their initial format (IIS, Common Log ...) to the Combined Log Format and filtered so as to keep only the meaningful web documents requests (such as html, pdf, asp, jsp file extensions). Then, the web user sessions have been built:

1. By identifying a web user by a triple "IP number, identifier (generally unknown) and user-agent".
2. By sorting the request by user and by date and,
3. According to an idle time of 30 minutes between two requests from the same identified user.

It is possible to improve this very simple method by adding knowledge from the web sites structure, to define page views instead of manipulating basic urls, but at the time of the study we do not have the complete information to deal with this problem. This task can become very complicated when working with dynamic web sites. This limitation of our preprocessing is not crucial since it is possible to recognize some of the frames that compose a single page view, during the interpretation of the results.

We also apply a filtering on the IP number and web user agent to detect and delete as much as possible web bots interactions with the web site. The detection of robots is crucial since our objective is to characterize web user accesses on the web site and not web bots. Moreover, the bad detection of robots can lead to large web sessions that are transversal to several sections of the web site and thus can cause the merging of several smaller web user sessions clusters focused on these particular sections. Nevertheless, we do not generalize the application of reverse DNS to discover robot from their IP number expressed as octets.

Web Sessions Similarity

Many representations of web user sessions have been already proposed in the literature. Initially, web sessions

were coded as numerical vectors because data mining algorithms usually take numerical vectors as input that are easier to handle and compare. Thus, the first works on web usage mining used transactions vectors, which are Boolean vectors that indicate for each page of a web site if it has been accessed during the session or not. This kind of representation was mainly used as input for association rules extraction algorithms. The second most used representation is the hits vector that indicates how many times each web page was accessed during the session. One may also consider the cumulative time spent on each page. However, these representations did not seem to be able to capture all the richness or specificity of the user activity.

In (Heer & Chi 2001), the authors introduce a session representation that relies on a weighted sum of numerical Term Frequency / Inverse Document Frequency vectors representative of the content of each accessed page to carry more information about the user information needs. The drawback is that each web page has to be preprocessed to generate the "keywords vectors" needed to analyze the sessions.

Other works represent web sessions as web pages or events sequences and introduce adapted similarity measures. In (Wang & Zaiane 2002), the authors propose a new similarity measure between web sessions inspired by existing sequence alignment algorithms. The method relies on a mechanism more or less similar to those of an edit distance between characters strings, with a cost associated with each basic sequence operation (insertion, suppression or substitution) needed to transform the first sequence into the second.

These edition costs are calculated according to a function of the similarity between the urls that compose the sequences. This similarity measure between urls doesn't rely on the content of the pages but on the path to access the documents as in (Nasraoui, Joshi, & Krishnapuram 1999).

Let p denotes the path of a url and let $|p|$ be the number of tokens in this path (its length). To compute the similarity between two urls u_1 and u_2, the method first assigns a weight to each token of each url. The first token is assigned the value ω that is defined as follows:

$$\omega = max(|pu_1|, |pu_2|) \tag{2}$$

where $|pu_1|$ (resp. $|pu_2|$) denotes the number of tokens in url u_1 (resp. u_2). The second token is set to the weight $\omega - 1$ and so on, until the last token of the longest path that is set to a weight equal to 1. The similarity $S_{url}(u_1, u_2)$ between urls u_1 and u_2 is computed as a sum over the first common tokens (possibly none) until a token differs from one url to the other. This sum value is then normalized by the sum of all weights from 1 to ω. The author apply their similarity measure on real web sessions but do not detail the results in term of quality of discovered clusters, and do not explain clearly the cost settings in their similarity measure between web sessions.

Other researches, on events sequences comparison, have been conducted and also rely on adapted edit-distance to take into account the order and the context of the events (Mannila & Ronkainen 1993).

We introduce in this section two similarity measures between web sessions. The first similarity measure is derived from the classical Levenshtein edit-distance that estimates the distance between a source string s and a target string t by evaluating the optimal number of characters deletions, insertions and substitutions that are needed to produce t starting from s according to the cost of each of these operations. In our application, where we consider web pages sequences, the substitution cost is computed according to the similarity between the considered web pages as described previously. If web pages are similar (respectively very different) the substitution cost should be near 0 (respectively near 1).

The following equation links the substitution cost $C_s(u_1, u_2)$ between two urls $u_{1,2}$ and the similarity $S_{url}(u_1, u_2)$ between urls:

$$C_s(u_1, u_2) = 1 - S_{url}(u_1, u_2) \tag{3}$$

The cost for insertion and deletion are set to 1 as we can consider these operations as the worst substitution case (i.e. to substitute a url with the empty url). The optimal cost $L_d(s_1, s_2)$ to obtain the target session s_2 from s_1 is determined by the Levenshtein edit-distance algorithm according to the previous costs. Thus, we define a normalized similarity $S_{seq}(s_1, s_2)$ between the sessions s_1 and s_2 as the following equation shows:

$$S_{seq}(s_1, s_2) = 1 - \frac{L_d(s_1, s_2)}{max(|s_1|, |s_2|)} \tag{4}$$

where $|s_1|$ (resp. $|s_2|$) is the length of the sequence s_1 (resp. s_2).

The second similarity measure is also based on the similarity between the urls of each session but does not take into account the order in which each url is accessed. Our similarity measure $S_{set}(s_1, s_2)$ between web sessions s_1 and s_2 simply estimates the mean url similarity between each url of s_1 and each url of s_2 as the following equation shows:

$$S_{set}(s_1, s_2) = \frac{\sum_{\forall u_1 \in s_1} \sum_{\forall u_2 \in s_2} S_{url}(u_1, u_2)}{|s_1| \times |s_2|} \tag{5}$$

Our similarity measures can deal with a large number of urls because they only rely on the comparison of the subset of urls that were accessed during the compared sessions and not the set of all urls. The only case where two web users who don't exhibit the same behaviour can be considered similar with our measures is when the structure of the web site is very simple (one main directory for example). In this case, the "syntactic" similarity between urls is greater than 0 and

so is the global similarity between sessions. This may cause some clusters to be larger than expected when web sessions access different parts of a directory on a web site.

Convergence of the Algorithms on Web Log Files

The clustering algorithms presented in this paper either heavily depend on their initialization (FCMdd or our Variant of Fuzzy C Medoid - VFCM) or are non-deterministic (leader Ant algorithm). Thus, we propose in this section, a brief analysis of the convergence of these algorithms.

As said previously, there are several ways to initialize the FCM-like algorithms. We use one of the initialization processes described in (Nasraoui 2002) that is deterministic. Thus, there is no convergence problem for our FCM-like algorithms as the output partition is always the same.

As the leader Ant algorithm is non-deterministic, we conducted experiments to compare partitions obtained on the 3 web sites. We cluster 50 times the sessions of each web site using the unordered set representation of web sessions and each time we record the computed partition. Then, each partition was compared to the others according to the similarity measure adapted from the measure developed by Fowlkes and Mallows as used in (Heer 2002). It evaluates the differences between two partitions by comparing each pair of objects and by verifying each time if they are clustered similarly or not. Let P_{exp} be the expected partition and P_{out} the output partition of a clustering algorithm. The clustering error $Err(P_{exp}, P_{out})$ can be defined as follows:

$$Err(P_{exp}, P_{out}) = \frac{2 \times \sum_{(a,b)\in[1,n]^2, a<b} \epsilon(a,b)}{n \times (n-1)} \quad (6)$$

where n is the number of objects in the data set, $P_{exp}(i)$ (resp. $P_{out}(i)$) denotes the index of the cluster of object i in the partition P_{exp} (resp. P_{out}), a and b are indices of objects and $\epsilon(a,b)$ is defined as follows:

- if a and b are clustered similarly (in the same or in distinct clusters) in both partitions P_{exp} and P_{out}, then $\epsilon(a,b) = 0$;
- else $\epsilon(a,b) = 1$.

The following table 1 introduces for each web site, the mean error between its partitions over 50 runs.

Web Site	Mean Error (50 runs)
Museum of Bourges	0.06
Cap Sciences	0.17
CCSTI Grenoble	0.03

Table 1: Mean convergence error over 50 runs on each web site.

These results show that although non-deterministic, the Leader Ant algorithm is able to produce very similar partitions over multiple runs on the same web data set. The errors for the Museum of Bourges and CCSTI Grenoble are close to 0, which indicates that the clustering is easier than the clustering of the Cap Sciences web sessions. This result will be confirmed by the analysis of the separability and compacity of the discovered clusters introduced in the following section.

Comparison of Similarity Measures

This section presents the experimental clustering results obtained when applying the two previous measures of similarity with the leader Ant algorithm on web log files recorded during one month and a half on the French museum "Cap Sciences" web site. As the objective of our approach is to characterize the accesses of the web users on the web site and to extract the main trends in their navigation, we propose for each discovered cluster a description based on the web pages that were accessed in the session of this cluster.

Description of Cap Sciences Web Site: the Cap Sciences web site is a complex web site to analyze since it is mainly built as a portal that leads to several interconnected web sites that share similar interests. For example, the two main parts of the web site are "Cap Sciences" and "Info Sciences". The first aims at promoting the exhibitions of the museum whereas the second is interested in scientific popularization.

On the one hand, due to its web portal construction, the urls are well structured in folders and sub folders in the web server file system, which may help our similarity measures to be efficient. On the other hand, some distinct urls lead to the same content, which introduces some noise and thus complicates the interpretation of clustering results according to the web pages that were accessed. The web log files were recorded during March, April and May 2005.

The tables 2 and 3 hereafter propose a description of the clusters based on the 20 most visited urls in each cluster for each similarity measure. The columns indicate for each cluster its number ($\#Cluster$), its number of sessions ($\#Sessions$) and a brief description of its content.

The first approach that handles web sessions as web pages sequences produces 17 clusters whose size vary from 740 to 5868 sessions. All the clusters are well defined as they generally only contain requests for one or two parts of the Cap Sciences web site. For example, the sessions in the clusters 1, 2, 3, 5, 6, 11, 15 only focus on "tourism" or a particular events or specific information available. On the one hand, that can lead us to think that some of web users of Cap Sciences directly browse to the section of interest with little interaction with the home page that contains navigational links. On the other hand, other clusters contain accesses to search pages and requests to the web site resource (as in cluster 10 and 17 as opposed to cluster 2). Several clusters deal with the download of teaching documents (cluster 2, 10, 13 and 17) and some of them contain requests for the same web pages. This problem is due to the similarity measure, which considers that two sessions must have visited similar urls in the same order to be similar.

# Cluster	# Sessions	Content
1	3943	Industry, academic training
2	5868	Teaching, documents download
3	1793	Events: sciences fest
4	1943	Cap Sciences (65%), Info Sciences index page, calendar, location
5	1613	Pictures and Networks
6	1015	Chemistry
7	740	Default page, calendar, location
8	3444	InfoSciences
9	3271	CapSciences home, topic "Aquitaine"
10	1511	Teaching docs download, search for documents
11	2364	Tourism
12	1792	Other CapSciences
13	3567	Teaching docs download
14	1197	Other InfoSciences
15	1578	Research docs download
16	1259	Other CapSciences
17	2106	Teaching docs download, other search for documents

Table 2: Content of the clusters discovered with the session representation as sequences.

# Cluster	# Sessions	Content
1	16756	Teaching docs download (90%), CapSciences home
2	4531	CapSciences home (80%)
3	2468	Tourism
4	1507	CapSciences home, tourism, industry, others
5	754	CapSciences home, industry, academic training
6	1608	Pictures and Networks
7	2163	Industry, teaching docs on "electricity"
8	1987	Cap Sciences home, calendar, location
9	2377	Cap Sciences home, events: sciences fest
10	2667	Cap Sciences home, others
11	1251	Other Cap Sciences
12	935	Chemitry

Table 3: Content of the clusters discovered with the session representation as unordered set of web pages.

The second approach represents sessions as a set of unordered urls and thus is less selective than the first approach and produces only 12 clusters (with the same clustering settings). The clusters are also well defined but contain a higher proportion of the home page of Cap Sciences in the 20 most requested urls. This is mainly due to the fact that as the similarity criteria is less selective, the clusters are bigger (from 754 to 16756 sessions) and thus add some noise with the home pages that are generally accessed by most of the users and thus rank easily in the 20 most visited pages.

The table 4 indicates for each similarity measure, the details of the running times of the clustering algorithm: the time spent for urls comparisons and then the global clustering time (in seconds). The experiments are conducted on a 1GHz ultra low voltage Intel Centrino with 1.28Go RAM and the algorithms are written in Java.

As table 4 shows, the second approach is approximately two times faster than the first one. Our experiments show

Method	Sequence	Unordered Set
Urls comparisons	$377.64s.$	$375.38s.$
Clustering time	$929.03s.$	$456.23s.$

Table 4: Running times with both similarity measures.

that each similarity measure has advantages and associated drawbacks: the representation of web sessions as a sequence of web pages produces clusters that are more "focused" and with less noise (i.e. accesses on the home pages of web sites) contrary to the other approach but can lead to too many clusters in some cases (as for the "teaching documents download" if they are not requested in the same order). The second measure does not take the order of web pages visits into account and thus is less selective than the first method. Consequently, it produces fewer clusters, with less computation but outputs slightly more noisy clusters.

Both approaches are efficient and manage to discover well-defined clusters that allow to observe the major navigation trends on the web site at a glimpse. According to the computation times, we propose to use the second similarity measure that is based on a representation of web sessions as an unordered set of web pages. The next section describes the results obtained with this similarity measure on the 3 web sites and introduces a quality measure of the discovered partitions.

Comparative Results of the Clustering Algorithms

This section presents the profiles discovered for each web site by each relational clustering algorithm presented in this paper.

The table 5 introduces for each web site the following information: the size of the web log files ($size$), the number of requests in these log files ($requests$), the number of sessions ($sessions$), the number of single sessions ($single$) that are generated by a user that only browse the web site once, the number of distinct urls ($urls$), and the mean number of requests per session ($meanreq.$).

	Bo	**Cs**	**Cg**
$size$	2.13 Mo	1.44 Go	5.5 Mo
$requests$	9913	-	28317
$sessions$	1512	24817	7407
$single$	932	18143	4772
$urls$	163	1174	896
$meanreq.$	6	3	3

Table 5: Information for Museum of Bourges (**Bo**), Cap Sciences (**Cs**) and CCSTI of Grenoble (**Cg**).

The museum of Bourges is a relatively small web site for which we analyzed less than 10000 requests. Although it is a museum that deals with every aspects of nature and living systems, its web site is specialized in the study and protection of bats. Thus, the index of the web site mainly deals with this subject. The other particularity of the web site is that its html content files are mainly located in the same directory, which does not facilitate the discrimination process of our similarity measure between sessions that relies on the path of accessed web pages. However, the leader Ant discovers 6 clusters from which we can clearly draw the following user profiles:

- users interested in the actuality section of the web site. In this case, two clusters differentiate users that access the actuality web pages through the index page and those that may be already used to the web site;
- users browsing exclusively the English content of the web site.

The LFCMdd algorithm was initialized with 5, 7 and 10 clusters. Each time, it produces a main cluster that gathers more than 1300 sessions (over 1512) and smaller clusters. Like the leader Ant, LFCMdd also discovers the groups of users interested in English content exclusively and actuality. With 10 clusters, LFCMdd mines very small clusters that contains from 1 to 18 sessions and that deals with English pages access via the home page of the web site, a hacker attack on the web server (one session), a direct access to the review paper pages (5 sessions) and users interested in animal species and collections.

The VFCM algorithm was also initialized with 5, 7 and 10 clusters. It produces clusters very similar to LFCMdd with the difference that even when it is set with 10 clusters, it builds a partition with only 7 clusters, two of them being very small (1 session each). One of the interesting clusters relative to the topic "actuality" - that is discovered by the other approaches - is hidden in the main cluster. The reason may be that our approach allows several clusters to have the same medoid, which in turn reduces the number of clusters and their interpretability.

These results are difficult to analyze since many web bots (that could not have been correctly filtered in the preprocessing step) accessed the web site (mainly the English content). These web bots may be the cause of the mean length of the sessions (6 requests, see table 5).

Our analysis of Cap Sciences web site is realized over three months to observe the evolution of the discovered profiles from one month to the next. Here are the main profiles that are extracted from the web log files:

- typical user accesses to a Museum web site to retrieve museum visit information (index page, map, calendar ...) is observed each month .This group of users is generally the more important counting more than 1000 sessions each month.
- users that visit the Cap Sciences web site to download documents in pdf format to prepare their visit or to obtain information on scientific subjects. These users do not necessarily visit the index page, which means they may use bookmarks or search engines. Further analysis should be conducted to conclude on this point.

In addition, some other groups of users can be found each month such as: "Pictures and Networks", "Chemistry and Info Sciences", "Sciences Fest" and other events. The LFCMdd and VFCM algorithms are initialized with 20 clusters and have been applied successively on each month of activity of Cap Sciences web site.

LFCMdd produces 20 clusters but 2 of them only contain 1 session for one month, which may indicate that this number is overestimated. LFCMdd discovers the same trends in the web users navigations as those enlighten by the leader Ant: clusters about general information about Cap Sciences institution, "Pictures and Networks", "Chemistry", "Sciences Fest" and the download of documents. VFCM produces respectively 11, 14 and 8 clusters for the 3 months of the study. The first two months, VFCM produces clusters that are comparable to those of LFCMdd but that contains much noise as they represent the same information with

less cluster. The interpretation is very difficult for the last month where all the information relies in the first cluster (that contains 2842 sessions).

The analysis of the evolution of clusters each month shows that some clusters exist each month (general information), some other are more and more represented over time (documents download) and some other tend to disappear because they are linked to a specific past actuality (exhibitions).

Concerning the web users profiles learning, the interesting points in this analysis are that:

- web users are mainly people that prepare a visit of the museum by accessing venue information
- the museum web site provides scientific and technological documents and thus plays an important role in sciences popularization, which was not expected at the beginning of the study.

The CCSTI of Grenoble web site promotes a large number of exhibitions and thus its index page contains many links and menus. The web log files cover 4 months from April to July 2005.

The clusters discovered by the leader Ant are mainly related to one exhibition or a part of the web site accessible from a menu, which helps understanding the underlying motivations of web users. 5 clusters are related to exhibitions accessible from the same menu in the center of the index page. The web users only access one exhibition at a time, which facilitates the interpretation of the clusters. 3 other clusters contain requests on general information pages about the CCSTI Grenoble and are not interested in the exhibitions. Finally, the other clusters are also well defined and contain requests for national or local events.

The LFCMdd and VFCM algorithms are set with 20 clusters and produce almost the same partition (mainly because they use the same initialization method). The clusters are well-defined and generally only access a portion of the web site (related to an exposition or an event), like with the leader Ant. The main cluster contains more than 3500 sessions with both approaches, but LFCMdd seems to build a better partition than VFCM: for a given event it produces a bigger cluster than VFCM and thus has a smaller and less noisy main cluster.

Quality of the Partitions of the Web Users Sessions

In order to be able to evaluate the quality of these results, we propose an aggregated estimator E of the compacity and separability of the discovered clusters. The estimator E is defined over all the couples of sessions as the ratio between the distance intra cluster D_{intra} and the distance inter cluster D_{inter}. The former D_{intra} is computed as the mean distance between sessions in the same clusters. The later D_{inter} is defined as the mean distance between sessions in different clusters. It can be formalized as follows:

$$E = \frac{D_{intra}}{D_{inter}} \quad (7)$$

$$D_{intra} = \frac{\sum_{\forall(i,j)\in S^2, i<j} \delta_{ci=cj} \times distance(i,j)}{\sum_{\forall(i,j)\in S^2, i<j} \delta_{ci=cj}} \quad (8)$$

$$D_{inter} = \frac{\sum_{\forall(i,j)\in S^2, i<j} \delta_{ci\neq cj} \times distance(i,j)}{\sum_{\forall(i,j)\in S^2, i<j} \delta_{ci\neq cj}} \quad (9)$$

where S is the set of n sessions, ci (resp. cj) is the cluster of session i (resp. j), δ is the identity function and $distance(i,j)$ is the distance value between session i and session j. One expect a clustering algorithm to produce E values smaller than 1, i.e. that minimizes the distance intra cluster and maximizes the distance inter clusters.

The table 6 indicates for each web site the quality of the partition computed over 50 runs on a subset or the totality of the web sessions available depending of their number.

E	**Bo**	**Cs**	**Cg**
Leader Ant	0.61 ± 0.02	0.88 ± 0.03	0.38 ± 0.03
LFCMdd	0.71 ± 0.00	0.94 ± 0.00	0.64 ± 0.00
VFCM	0.72 ± 0.00	0.98 ± 0.00	0.67 ± 0.00

Table 6: Mean quality E of the partitions and its standard deviation over 50 runs for the three clustering algorithms and three web sites: museum of Bourges (**Bo**), Cap Sciences (**Cs**) and CCSTI of Grenoble (**Cg**).

As the table 6 shows, all the partitions discovered have a quality value E below 1, which indicates that the clustering algorithm manages to optimize the distances intra and inter clusters. However, we can notice that there are differences in the results for the three web sites and three clustering algorithms, which can denotes distinct clustering difficulties.

As expected from the previous experiment, the Cap Sciences web site has the maximum E value for all clustering algorithms. These results confirm that this web site is the most difficult to analyze for the algorithms, and that they produce clusters with little noise. The leader Ant has the lowest score of 0.88, which tends to show that its partition may be the easiest to interpret. In fact, the values of $D_{intra} = 0.86$ and $D_{inter} = 0.99$ indicate that although the clusters are well separated, they are not as compact as expected. This situation is worst for LFCMdd ($D_{intra} = 0.94$, $D_{inter} = 0.99$) and VFCM ($D_{intra} = 0.96$, $D_{inter} = 0.98$) that produces clusters in which objects within a cluster are almost as similar as with objects from other clusters.

The clustering of the museum of Bourges web sessions has a better E value, from 0.61 (leader Ant) to 0.72 (VFCM). These values can be explained by the fact that the clusters are more clearly defined than for the Cap Sciences web site, with accesses to distinct part of the web site. The analysis

gives $D_{intra} = 0.55$ for the leader Ant, $D_{intra} = 0.65$ for VFCM and $D_{intra} = 0.63$ for LFCMdd, which indicates that the clusters are much more compact than for Cap Sciences web site and as well separated ($D_{inter} = 0.90$).

Finally, the CCSTI Grenoble has the smallest E value for all the clustering algorithms. The leader Ant produces the best partition with a score of 0.38. This can be explained by the large number of clusters which are very well defined (mainly focused on a single exhibition) which is confirmed by a small distance intra cluster for all clustering algorithms (leader Ant: $D_{intra} = 0.38$; LFCMdd: $D_{intra} = 0.63$; VFCM: $D_{intra} = 0.66$) and a large distance inter cluster ($D_{inter} = 0.99$).

Conclusion and Perspectives

This paper proposes a comparison of relational clustering algorithms on web usage data to characterize user access profiles. These methods only rely on numerical values that represents the distance or the dissimilarity between web user sessions to construct web user profiles (and not the sessions themselves), and thus are useful in the context of web personalization.

We describe two relational clustering algorithm: the leader Ant clustering algorithm that is inspired by the chemical recognition system of ants and a new variant of traditional fuzzy C medoids named VFCM. We compare these algorithms to the well-known LFCMdd algorithm (Krishnapuram *et al.* 2001). We also describe two similarity measures that handle web sessions either as a sequence or a set of urls.

The paper describes the application of these clustering algorithms on real web usage data sets from French museums. The evaluation is conducted according to the quality of the output partitions and the interpretability of each cluster based on its content.

Although simple, the leader Ant shows that it can produce cluster that are more compact and more separated than the other approaches. Moreover, it allows to cluster data in linear time with the number of objects, does not need to be parameterized with a number of clusters and, as any relational clustering algorithm, can be associated with any similarity measure that outputs a value between 0 and 1. In this model, the random encounters between artificial ants replace the mean or medoid computation of k-Means like approaches and thus allow to compare non numerical data such as web sessions. LFCMdd and VFCM have also good results and have proven to be more efficient on numerical benchmark data sets, once the expected number of clusters provided to the algorithms. The advantage of these approaches is that they build prototypes that may be directly used as a basis for personalization, which is not (yet) the case for the ant based method.

Although, this paper only reports three experiments on museums web sites, the leader Ant algorithm has been applied on five other web sites related to sciences and technologies with success. Our study allow to enlighten some typical web user profiles of the museums:

- Users that search information on the museum by browsing general information pages (index, map, schedules). Although expected, this result is confirmed by our study.
- Users that access the web site to download documents. This result is very interesting, because it was not expected. Some museum web sites play an important role as scientific documents providers, which complete the information available in the exhibitions of the museums. In some clusters, users only access download documents such as pdf files.
- Users that browse the job opportunities web pages or search information on specific research team, as some of the institutions are also large French research centers.

In our future work, we plan to develop incremental relational clustering algorithms that can handle the continuous flow of users navigations on web servers so as to always have up to date users profiles. We also begin to work on the problem of web user navigation visualization and clustering in an adaptive and interactive framework to help the webmaster understanding web user navigations.

References

Baraglia, R., and Palmerini, P. 2002. Suggest: A web usage mining system. In *Proceedings of IEEE International Conference on Information Technology: Coding and Computing*.

Bezdek, J. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Cap Sciences. 2007. http://www.cap-sciences.net.

CCSTI Grenoble. 2007. http://www.ccsti-grenoble.org/.

Cooley, R.; Mobasher, B.; and Srivastava, J. 1999. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems* 1(1):5–32.

Cooley, R.; Srivastava, J.; and Mobasher, B. 1997. Web mining: Information and pattern discovery on the world wide web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*.

Davison, B. 1999. Adaptive web prefetching. In *Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*, 105–106.

Davison, B. 2002. Predicting web actions from html content. In *Proceedings of the The Thirteenth ACM Conference on Hypertext and Hypermedia (HT'02)*, 159–168.

Estivill-Castro, V., and Yang, J. 2001. Categorizing visitors dynamically by fast and robust clustering of access logs. *Lecture Notes in Computer Science* 2198:498–509.

Fu, Y.; Sandhu, K.; and Shih, M. 1999. Clustering of web users based on access patterns. In Springer-Verlag., ed., *Proceedings of the 1999 KDD Workshop on Web Mining*.

Heer, J., and Chi, E. 2001. Identification of web user traffic composition using multi-modal clustering and information

scent. In *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining*, 51–58.

Heer, J., and Chi, E. 2002. Mining the structure of user activity using cluster stability. In *Proceedings of the Workshop on Web Analytics, SIAM Conference on Data Mining*.

Kay, J.; Maisonneuve, N.; Yacef, K.; and Zaiane, O. 2006. Mining patterns of events in studentsťeamwork data. In *Proceedings of the Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*, 45–52.

Krishnapuram, R.; Joshi, A.; Nasraoui, O.; and Yi, L. 2001. Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE-FS* 9:595–607.

Krishnapuram, R.; Joshi, A.; and Yi, L. 1999. A fuzzy relative of the k-medoids algorithm with application to document and snippet clustering. In *Proceedings of IEEE Intl. Conf. Fuzzy Systems - FUZZIEEE 99*.

Labroche, N.; Monmarché, N.; and Venturini, G. 2003. Antclust: Ant clustering and web usage mining. In *Proceedings of the Genetic and Evolutionary Computation Conference (Gecco 2003)*.

Labroche, N. 2006a. Clustering web pages sequences with artificial ants. In *IADIS International WWW/Internet Conference*, 503–510.

Labroche, N. 2006b. Fast ant-inspired clustering algorithm for web usage mining. In *IPMU 2006 Conference*, 2668–2675.

Mannila, H., and Ronkainen, P. 1993. Similarity of event sequences (extended abstract).

Masseglia, F.; Poncelet, P.; and Cicchetti, R. 1999. Webtool: An integrated framework for data mining. In *Database and Expert Systems Applications*, 892–901.

Masseglia, F.; Poncelet, P.; and Teisseire, M. 1999. Using data mining techniques on web access logs to dynamically improve hypertext structure. *ACM SigWeb Letters* 8(3):1–19.

Mobasher, B. 2006. *The Adaptive Web: Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science, Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.)*. Springer-Verlag, Berlin Heidelberg. chapter Data Mining for Personalization.

Museum of Bourges. 2007. http://www.museum-bourges.net/anglais/.

Nasraoui, O.; Frigui, H.; Joshi, A.; and Krishnapuram, R. 1999. Mining web access logs using relational competitive fuzzy clustering. In *Eight International Fuzzy Systems Association World Congress - IFSA 99*.

Nasraoui, O.; Krishnapuram, R.; Joshi, A.; and Kamdar, T. 2002. *Automatic Web User Profiling and Personalization using Robust Fuzzy Relational Clustering*. E-Commerce and Intelligent Methods in the series Studies in Fuzziness and Soft Computing, J. Kacprzyk, Ed. Springer-Verlag.

Nasraoui, O.; Joshi, A.; and Krishnapuram, R. 1999. Relational clustering based on a new robust estimator with application to web mining. In *Proc. of Intl. Conf. North American Fuzzy Info. Proc. Society (NAFIPS 99)*.

Perkowitz, M., and Etzioni, O. 1999. Adaptive web sites: Conceptual cluster mining. In *Sixteenth International Joint Conference on Articial Intelligence*, 264–269.

Spiliopoulou, M., and Faulstich, L. 1998. Wum: a web utilization miner. In *Workshop on the Web and Data Bases (WebDB98)*, 109–115.

Spiliopoulou, M.; Pohle, C.; and Faulstich, L. 1999. Improving the effectiveness of a web site with web usage mining. In *Proceedings of the WebKDD Conference*, 142–162.

Suryavanshi, B.; Shiri, N.; and Mudur, S. 2005. Incremental relational fuzzy subtractive clustering for dynamic web usage profiling. In *Proc. WEBKDD Workshop on Taming Evolving, Expanding and Multi-faceted Web Clickstreams*.

Wang, W., and Zaiane, O. 2002. Clustering web sessions by sequence alignment. In *3rd International Workshop on Management of Information on the Web in conjunction with DEXA2002*, 394–398.

Yan, T.; Jacobsen, M.; Garcia-Molina, H.; and Dayal, U. 1996. From user access patterns to dynamic hypertext linking. In *Proc. of 5th WWW*, 1007–1014.