Web Personalization Based on Static Information and Dynamic User Behavior[*]

Massimiliano Albanese
malbanes@unina.it

Antonio Picariello
picus@unina.it

Carlo and Lucio Sansone
{carlosan,sansone}@unina.it

Dipartimento di Informatica e Sistemistica
Università di Napoli “Federico II”
Napoli, Italy

ABSTRACT

The explosive growth of the web is at the basis of the great interest into web usage mining techniques in both commercial and research areas. In this paper, a web personalization strategy based on pattern recognition techniques is presented. This strategy takes into account both static information, by means of classical clustering algorithms, and dynamic behavior of a user, proposing a novel and effective re-classification algorithm. Experiments have been carried out in order to validate our approach and evaluate the proposed algorithm.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms, Experimentation

Keywords

Clustering, web personalization, web usage mining

1. INTRODUCTION

It is well known that the World Wide Web may be considered as a huge and global information center. A *web site* usually contains a great amount of information distributed through hundreds of pages. Without proper guidance, a visitor often wanders aimlessly without visiting important pages, loses interest and leaves the site sooner than expected. This consideration is at the basis of the great interest in web mining both in the academic and the industrial world.

Usually, three types of data have to be managed in a web site: *content*, *structure* and *log* data. *Content data* consist of whatever is in a web page; *structure data* refer to the organization of the content; *usage data* are the usage patterns of web sites. The application of data mining techniques to these different data sets is at the basis of the three different research directions in the field of web mining: *web content mining*, *web structure mining* and *web usage mining* [18].

In this paper, we are interested in the *web usage mining* domain, which is usually described as *the process of customizing the content and the structure of web sites* in order to provide users with the information they are interested in, without asking for it explicitly [4, 11]. Various personalization schemes have been suggested in the literature. *Letizia* [9] is perhaps the first system which takes into account the user's navigation through a web site. This goal is achieved by using a client-side agent that records the user's behavior and gives interesting recommendations to the user herself. Yan et al. [16] propose a methodology for the automatic classification of web users according to their access patterns, using cluster analysis on the web logs. In [8], Joachims et al. describe *WebWatcher*, and similarly the *Personal WebWatcher* in [10], an intelligent agent system that provides navigation hints to the user, on the basis of a knowledge of the user's interests, the location and relevance of the many items in the site, and the way in which other users interacted with the collection in the past.

In the SpeedTracer project, Wu et al. [14] use statistically dominant paths and association rules discovery, previously developed by Chen et al. [3]: each user session is mapped into a transaction and then data mining techniques are applied in order to discover the most frequent user traversal paths and the most frequently visited groups of pages. Zaiane et al. [17] and similarly Huang et al. [6] propose the use of cube models to extract knowledge about the user behavior. Similarly, Buchner and Mulvenna [1] describe a knowledge discovery system which combines existing online analytical mining and marketing expertise. Very important is also the paper of Perkowitz and Etzioni [12], that first describes adaptive web sites as sites that semiautomatically improve their organization by learning from visitor access patterns. They used an algorithm (*PageGather*) based on a clustering methodology. Srivastava et al. [13] have recently published a survey on the existing *web usage mining*

[*] This work has been carried out partially under the financial support of the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) in the framework of the FIRB Project “Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)”

WIDM'04, November 12–13, 2004, Washington, DC, USA.
Copyright 2004 ACM 1-58113-978-0/04/0011 ...$5.00.

projects. They also describe a prototype system (*WebSIFT*) which performs intelligent cleansing and preprocessing for identifying users. It infers page references through the use of the referrer field, and also performs content and structure preprocessing.

A great number of papers also deals with time-related issues. In [5] Grandi introduces an exhaustive annotated bibliography on temporal and evolution aspects in the World Wide Web. Several time-related issues have been investigated, among which we are primarily interested in *navigation time*, that can be defined as the temporal dimension marking the navigation of the Web by a user.

We can finally conclude that most of the existing works try to classify a user *i*) while she is browsing the web site or *ii*) using registration information. Our main criticism stands in the fact that in some applications it is not possible to perform an "on line" classification if the number of visited pages is not sufficiently great. By the way, using the registration forms alone may result inaccurate if the interests of a user change over time. The novelty of our approach is that of proposing a clustering process made up of two phases: in the first one a pattern analysis and classification is performed by means of an unsupervised clustering algorithm, using the registration information provided by the users. In the second one a re-classification is iteratively repeated until a suitable convergence is reached. Re-classification is used to overcome the inaccuracy of the registration information, based on the users' navigational behavior. To the best of our knowledge, this paper is the first one which uses re-classification in order to address both static and dynamic requirements.

The remainder of the paper is organized as follows: section 2 introduces the overall architecture of the proposed web personalization system, while section 3 describes our novel web usage mining strategy. A re-classification algorithm is proposed and described in section 3.2. Experiments and results are reported and discussed in section 4. Eventually, conclusions are reported in 5.

2. SYSTEM ARCHITECTURE

A web personalization system usually consists of the following modules, namely:

- *User profiling.* The process of gathering information specific to each visitor, either explicitly or implicitly. A user profile includes personal data about the user, her interests and behavior when browsing a web site.

- *Log analysis and web usage mining.* The process of analyzing the information stored in web server logs by means of data mining techniques, in order to (a) extract statistical information and discover interesting usage patterns, (b) cluster the users into groups according to their behavior, and (c) discover potential correlations between web pages and user groups.

- *Content management.* The process of classifying the content of a web site into semantic categories in order to make information retrieval and presentation easier for the users. Content management is fundamental for web sites whose content is increasing on a daily basis, such as news sites or portals.

- *Web site publishing* [4]. A publishing mechanism that is used to present the content stored in the web server and/or information retrieved from other web resources in a uniform way to the end-user.

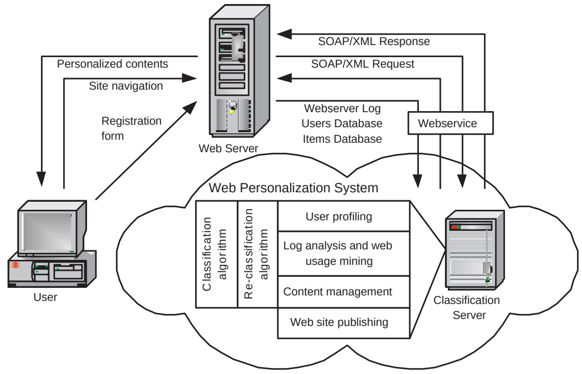

Figure 1: System architecture

In other words, the steps of a *web personalization* process are: (a) the collection of web data, (b) the modelling and categorization of these data (preprocessing), (c) the analysis of collected data, and (d) the determination of the actions that should be performed. The site is personalized through the highlighting of existing hyperlinks, the dynamic insertion of new hyperlinks that seem to be of interest for the current user, or even the creation of new index pages.

Figure 1 shows the overall architecture of the system. A standard client/server interaction occurs between the users and the *Web Server*. The algorithms for the classification and re-classification of the users are implemented on a distinct node of the network. Let us denote this node with the term *Web Personalization System*.

The *Web Server* and the *Web Personalization System* communicate via web services technology. Several advantages derive from the allocation of web site and classification system on two distinct nodes: (a) each node can be optimized for a particular task, improving the overall performances; (b) the personalization system may offer its services to more than a single web server, thus distributing the design and maintenance costs; (c) classification/re-classification algorithms can be modified in a transparent way for the web servers, if the web service interface is maintained.

3. THE WEB USAGE MINING STRATEGY

In this section we propose a novel web usage mining strategy. It requires users to subscribe and fill-in a registration form. As stated in the introduction, our approach is made up of two phases: in the first one the users are attributed to a tentative class by means of an unsupervised clustering algorithm. It uses the static information provided by the users themselves during the registration and also determines the number of classes the users can belong to. In the second phase, a novel re-classification algorithm is iteratively repeated until a suitable convergence in attributing each user to a class is reached. Re-classification is used to overcome the inaccuracy of the registration information and is accomplished by the log analysis and content management modules, on the basis of the dynamic navigational behavior of the users.

3.1 Pattern analysis, clustering and classification

The task of the *User profiling* module is to associate each user to the class that better describes her behavior. The problem of assigning a user to a particular class can be seen as a classical *pattern recognition* problem. To accomplish this task, users have to be mapped into a feature space. So, in a pre-processing stage, following the suggestions of a domain expert, a set of suitable features for describing web site users must be determined. Since we required users to register themselves, these features can be extracted from the information provided during the registration. In this case, a feature may be the value of a field in the registration form or a function of two or more fields.

The first issue of the user profiling task is the definition of the classes the users can belong to. Note that the number of classes can be easily determined only in a few cases (for example, in an educational site the users can be roughly classified as students, teachers or visitors). In most cases, indeed, there is too little knowledge about the domain at hand for establishing this number in a reliable way. So, an unsupervised clustering procedure can be used for partitioning the feature space into a certain number of clusters (each one representing a class) that group together users appearing to be similar within the chosen feature space. In order to choice the optimal number of clusters, say M, a possible approach could be the maximization of an index that measures the quality of the obtained clustering on a subset of the whole data set. A widely used index is the so-called C index [7], that is defined as follows:

$$\mathsf{C} = \frac{S - S_{min}}{S_{max} - S_{min}} \tag{1}$$

S is the sum of distances over all pairs of patterns from the same cluster. Let l be the number of those pairs; then S_{min} is the sum of the l smallest distances if all pairs of patterns are considered (i.e. if the patterns can belong to different clusters). Similarly S_{max} is the sum of the l largest distances out of all pairs. It is easy to see that the nominator in the formula will be small if pairs of patterns with a small distance are in the same cluster. Hence, a small value of C indicates a good clustering. So, we are interested in the minimization of the C index, as the number of clusters varies. Note that the denominator serves the purpose of normalization, causing $\mathsf{C} \in [0, 1]$.

This index can be also used for comparing different clustering techniques; the one that ensures the minimum value of C will be chosen. In this paper we propose and compare two different clustering techniques: AutoClass C [2], a fuzzy clustering algorithm based on the Bayesian theory, and the Rival Penalized Competitive Learning (RPCL) [15] algorithm, that is used for training a competitive neural network able to provide a crisp partitioning of the feature space.

In the first approach each cluster (class) is described by means of a likelihood function depending on some parameters. Given the number M of classes, the AutoClass C *Search* module estimates such parameters on the training data and finds the partition of the feature space that maximizes the log-likelihood value. The classification is performed by the *Prediction* module of AutoClass C: by using the Bayesian rule and the likelihood functions of each class, it attributes an object to each class with a different a posteriori probability; the predicted class is the one that exhibits the maximum a posteriori probability.

In the approach based on a competitive neural network, each neural unit represents, in the feature space, the centroid of a cluster: so a network with k units can be used for partitioning the feature space into k clusters. The RPCL learning algorithm uses a mechanism that demonstrated its ability of allocating just a neural unit for each cluster. Therefore, it is able to leave unused some neural units if their number is greater than the number of clusters. Once the training phase has been completed, the neural network can be used for classifying the users: it takes as input the feature vector x that represents a user and assign it to the cluster whose centroid has the smallest distance from x.

So, in both cases, the result of the clustering procedure is the initial class of each user. If a new user registers herself at the web site, she is classified according to the same schema. Finally, if a user explicitly changes the data in her registration form, she is classified again.

3.2 The re-classification algorithm

The re-classification phase is based on the interaction of each user with the web site and it is fundamental to make the personalization system robust against incomplete or erroneous information provided during the registration. The basic idea behind the proposed re-classification schema is to iteratively 1) classify the web site resources based on the type of users that have accessed them and 2) re-classify the users based on both the class they were previously assigned and the resources they have accessed since the last re-classification.

Without any loss of generality, we can suppose that the interaction can be realized in three different ways: *i*) by submitting queries containing some keywords, *ii*) by searching among directories or *iii*) by accessing pages that contain news or articles. So, three different resource types can be usually considered within a web site: queries, directories, and news/articles. It is worth noticing that the request of different resource types by a specific user provides different information about her real preferences or needs. For example, searching a directory that contains information about hotels can be considered as a less effective indicator of the user needs than the explicit request, provided by clicking on a specific link, of an article that contains information about a given hotel.

All the material on the web site is managed by the *Content management* module which, by means of a domain expert, selects all the significant keywords appearing into the different resources of the web site and associates them to a specific content category. This obviously implies that several keywords belong to the same content category.

On the other hand, the *Log analysis* module registers all the activities of the users. In order to use this information for re-classifying users we need to attribute each content category to a specific user class. Moreover, since the importance that can be associated to the use of the different resource types is typically different, we are interested in attributing each content category to a class depending on the specific resource type. This implies that each content category is separately attributed to (possibly) three different classes: a class within the query resource type, a class within the search resource type and a class within the news/article resource type.

```
procedure re-classification(in $C_{ij}(T_0)$, $T_l$, $N_j(T_{l-1})$, $W$; out $C_{ij}(T_l)$)
  $C_{ij}(T_0)$ is the initial classification produced by a clustering algorithm that attributes each user $u_i$ to a class $C_j$
  $T_l$ is the time at which the $l$-th re-classification is performed
  $N_j(T_{l-1})$ is the number of users belonging to each class $C_j$ after the previous re-classification[a]
  $W$ is a weight vector
  $C_{ij}(T_l)$ is the re-classification of all the users at the re-classification time $T_l$.
begin
  foreach resource_type $r \in \{queries, directories, news/articles\}$
    foreach content_category $CC^r_k$
      Count the number of times $n^r_{kj}(T_l)$ the users of each class $C_j$ have asked for a resource of type $r$ belonging to
        that content category, in the interval $[T_0, T_l]$
      Calculate the 'normalized request' $NR^r_{kj}(T_l)$ of that content category by the users of each class $C_j$ as
        $NR^r_{kj}(T_l) = \frac{n^r_{kj}(T_l)}{N_j(T_{l-1})}$
      Assign each content category $CC^r_k$ to the class $C_j$ with a probability $P^r_{kj}(T_l) = \frac{NR^r_{kj}(T_l)}{\sum_{\hat{j}} NR^r_{k\hat{j}}(T_l)}$
    end for
  end for
  foreach user $u_i$
    foreach resource_type $r \in \{queries, directories, news/articles\}$
      foreach content_category $CC^r_k$
        Count the number of times $nCC^r_{ki}(T_l)$ each user asked for that category in the interval $[T_0, T_l]$
      end for
      foreach class $C_j$
        Evaluate the quantity $P^r_{ij}(T_l) = \frac{\sum_k P^r_{kj}(T_l) \cdot nCC^r_{ki}(T_l)}{\sum_k nCC^r_{ki}(T_l)}$ . It represents the probability that the user $u_i$ belongs to the
          class $C_j$, since she asked for a given set of categories within a resource type $r$
      end for
    end for
    foreach class $C_j$
      Evaluate the quantity $P_{ij}(T_l) = C_{ij}(T_0) \cdot W^0 + \sum_r P^r_{ij}(T_l) \cdot W^r$, where the weight vector $W$ takes into account the
        different importance that can be associated to the use of the different resource types. In this case, $W^0$ is the
        weight associated to the initial classification performed by the chosen clustering algorithm and represents how we
        are confident in that classification
    end for
    Assign the user $u_i$ to the class $C_{ik}(T_l)$ such that $P_{ik}(T_l) = \max_j P_{ij}(T_l)$
  end for
end
```

[a] For the first re-classification, it is the number of users per class produced by the chosen clustering algorithm.

Figure 2: Re-classification algorithm

The process of attributing each category to a class can be accomplished by considering the first classification performed by the chosen clustering algorithm and by counting the number of times the users of a given class ask for something belonging to a specific content category within each resource type. Each content category can be, in fact, attributed to a class with a probability that is proportional to the number of requests made by the users of that class. This way of classifying the content categories can suffer the inaccuracy of the first classification. However, if the percentage of correctly classified users achieved by the chosen clustering algorithm is acceptable (say, greater than 50%) and the time interval T_l used for re-classifying users is long enough, the classification of the content categories can be considered reliable. A re-classification can be then performed, by considering the content categories requested by a user in a predefined time interval T_l. If the majority of the requests of a user refer to content categories belonging to a class other than her initial class, the user is re-classified. More precisely, the re-classification is performed by suitably weighting the initial class of each user and the probability she belongs to other classes, based on the content categories she asked for during the interval T_l.

The re-classification algorithm is described in more details in figure 2. Note that the time interval T_l increases at each re-classification, since it represents the interval between the initial classification and the current re-classification. The whole process of dynamically changing the class of each user will lead to convergence if after a suitable number of re-classifications the number of re-classified users leads to zero.

3.3 Web Personalization

Given the class of a user, the contents related to the categories attributed to that class will be shown by the web site publishing module on her personalized home page when she logs in the web site. For example, all the news and the stories containing keywords related to those categories will be presented, as well as the results of last queries involving the same keywords. Moreover, since each user is requested to register herself, different mailing lists related to the different content categories can be created, and specific mails can be sent to each user.

# clusters	C
2	0.511
3	0.393
4	0.362
5	0.378
6	0.307
7	**0.276**
8	0.283
9	0.284

Table 1: Evaluation of the clustering produced by AutoClass C

# neurons	C
2	0.237
3	0.150
4	0.125
5	0.117
6	0.108
7	**0.076**
8	0.082
9	0.085

Table 2: Evaluation of the clustering produced by RPCL

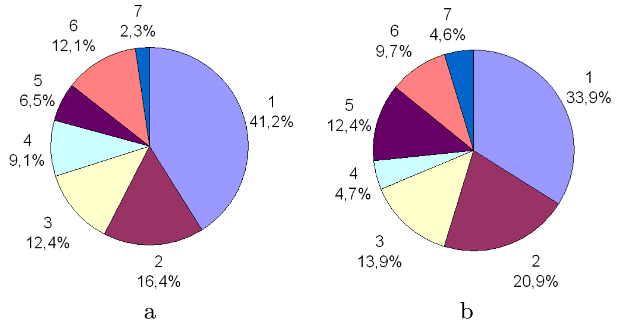

Figure 3: Initial distribution of users among classes produced by a) AutoClass C **b)** RPCL

4. EXPERIMENTAL RESULTS

In this section we report the experiments that have been carried out in order to validate the effectiveness of our approach and evaluate the proposed re-classification algorithm. A prototypal system has been implemented w.r.t. the architecture described in section 2. We chose an integrated framework for the deployment of web services, the Microsoft .NET framework. The web services implemented in the .NET framework use SOAP (Simple Object Application Protocol) for exchanging data between the web server and the personalization system. SOAP is a lightweight, XML based protocol for exchange of information in a decentralized, distributed environment.

As a case study, we have considered a commercial web site called pariare.com, which provides information about entertainment in the metropolitan area of Napoli (Italy). The web site, that is usually visited by hundreds of users a day, has been monitored for a period of six weeks. During this time interval the percentage of re-classified users has been tracked together with the percentages of transitions from a class to another one. The users already registered to the web site when the experimentation started have been initially classified using a standard clustering algorithm. Experiments have been repeated using each of the clustering algorithms described in section 3.1, in order to verify that the re-classification algorithm leads to convergence, whatever the initial classification is.

The data set used in the experiments consists of 2682 users. The features used to initially classify the users are: (1) age; (2) sex; (3) category of places in which users prefer to go; (4) number of times per week in which users go out; (5) preferred day of the week to go out; (5) the *Pariapoli* parameter (a measure of the degree of interest towards the virtual community of *Pariare*, evaluated as the normalized number of the information fields filled in the registration form); (7) type of entertainment users are looking for.

We have first determined the optimal number of clusters for classifying the users. In the literature several cluster validation indices have been proposed to measure the quality of a clustering. To the aim of this work we have adopted the C index described in section 3.1, that is easy to implement and has a low computational cost. Each of the two selected clustering algorithms has been executed several times, with a different number of clusters at each run, and the clustering that optimized the index has been selected as the final result.

Tables 1 and 2 report the evaluation of the clustering produced by AutoClass C and RPCL respectively for different values of the number of classes/neurons. In both cases the optimal number of clusters results to be 7. We have not considered a number of clusters higher than 9, because the value of the index monotonically increases when the number of clusters is higher than 7.

Figure 3 shows the initial distributions of users among classes produced by AutoClass C and RPCL respectively.

During the experimentation period the users of the web site have been re-classified several times, once every 10 days. Let us consider first the case of the initial classification performed by means of AutoClass C. The percentages of transition of a user from a class to another one, registered during the first re-classification, are shown in table 3.I. The generic element (i, j) of the transition matrix represents the percentage of i-class users who have been re-classified as j-class users and all the elements along a row sum up to 1. In particular an element (i, i) on the diagonal represents the percentage of i-class users which have not been re-classified. Ten days later, the re-classification algorithm was executed for the second time, thus obtaining the transition percentages reported in table 3.II. Let us observe that the percentages of non-re-classified users is quite close to 1, after only two runs of the re-classification algorithm. The overall percentage of re-classified users amounts to 3.14%.

Ten days later, the re-classification algorithm was executed for the third time, obtaining the transition percentages shown in table 3.III. We can observe that the values on the diagonal are very close to 1, so we expect that convergence will be reached within the next run of the algorithm. At this step, the percentage of re-classified users amounts to 0.62%. The re-classification process was executed for the last time other ten days later, producing the transition matrix in table 3.IV. The values on the diagonal are now equal or very close to 1. At this step, the percentage of re-classified users amounts to 0.16%. Figure 4 shows how the percentage of re-classified users converges towards zero.

		1	2	3	4	5	6	7
I	**1**	**0.774**	0.017	0.033	0.134	0.001	0.006	0.034
	2	0.028	**0.679**	0.055	0.173	0.000	0.006	0.059
	3	0.005	0.016	**0.879**	0.065	0.005	0.005	0.024
	4	0.011	0.000	0.040	**0.919**	0.000	0.007	0.022
	5	0.088	0.015	0.052	0.216	**0.572**	0.010	0.046
	6	0.060	0.027	0.082	0.165	0.000	**0.613**	0.052
	7	0.000	0.000	0.101	0.217	0.000	0.000	**0.681**
II	**1**	**0.970**	0.001	0.002	0.021	0.000	0.000	0.006
	2	0.000	**0.976**	0.000	0.021	0.000	0.000	0.003
	3	0.002	0.000	**0.980**	0.013	0.000	0.000	0.004
	4	0.006	0.000	0.003	**0.983**	0.000	0.005	0.003
	5	0.009	0.000	0.000	0.079	**0.833**	0.000	0.079
	6	0.000	0.000	0.000	0.054	0.000	**0.942**	0.004
	7	0.000	0.000	0.000	0.012	0.000	0.000	**0.988**
III	**1**	**0.994**	0.000	0.002	0.003	0.000	0.000	0.001
	2	0.000	**0.995**	0.000	0.005	0.000	0.000	0.000
	3	0.000	0.000	**0.996**	0.002	0.000	0.000	0.002
	4	0.004	0.000	0.001	**0.993**	0.000	0.001	0.000
	5	0.000	0.000	0.000	0.010	**0.990**	0.000	0.000
	6	0.004	0.000	0.000	0.004	0.000	**0.992**	0.000
	7	0.000	0.000	0.000	0.006	0.000	0.000	**0.994**
IV	**1**	**0.998**	0.000	0.000	0.002	0.000	0.000	0.000
	2	0.000	**1.000**	0.000	0.000	0.000	0.000	0.000
	3	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000
	4	0.000	0.000	0.001	**0.999**	0.000	0.000	0.000
	5	0.000	0.000	0.000	0.010	**0.990**	0.000	0.000
	6	0.000	0.000	0.000	0.004	0.000	**0.996**	0.000
	7	0.000	0.000	0.000	0.000	0.000	0.000	**1.000**

Table 3: Transition percentages produced by the re-classification algorithm w.r.t. AutoClass C clustering

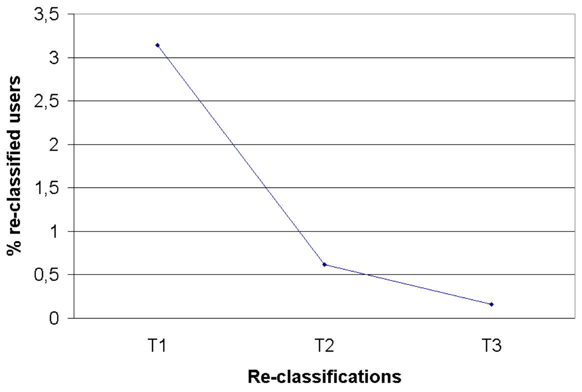

Figure 4: Percentage of re-classified users vs time w.r.t AutoClass C clustering

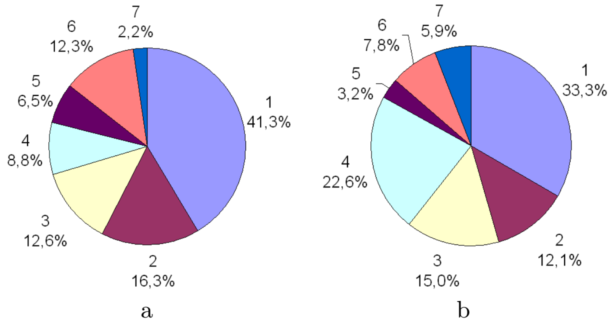

Figure 5: Distribution of users among classes produced a) by AutoClass C at the time the last re-classification; b) by the last run of the re-classification algorithm

Figure 5.a shows the distribution of the users among the classes produced by AutoClass C on the same day of the last re-classification[1], while figure 5.b shows the distribution of the users among the classes after the last re-classification. The first one is based only on the data provided by the users in the registration form, while the last one has been determined using both static and dynamic information. The comparison between the two distribution shows the benefits of adopting a classification strategy that takes into account both the data provided during the registration and the navigational behavior of the users: such a strategy can best fit changes in the behavior of the users.

In a similar way we have analyzed the convergence of the re-classification algorithm, starting from the initial classification produced by RPCL. Table 4 reports the transition matrix determined during the successive re-classifications.

Also in this case the re-classification process leads to a stable classification in a few runs of the re-classification al-

[1] Difference w.r.t. the initial classification are due to new registered users.

		1	2	3	4	5	6	7
I	**1**	**0.863**	0.005	0.010	0.063	0.010	0.003	0.046
	2	0.000	**0.681**	0.006	0.086	0.037	0.006	0.184
	3	0.008	0.008	**0.724**	0.089	0.014	0.000	0.158
	4	0.002	0.005	0.012	**0.824**	0.005	0.002	0.149
	5	0.000	0.010	0.012	0.060	**0.806**	0.002	0.109
	6	0.012	0.012	0.004	0.099	0.006	**0.680**	0.193
	7	0.004	0.004	0.007	0.046	0.007	0.006	**0.926**
II	**1**	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.031	**0.922**	0.000	0.039	0.000	0.000	0.008
	3	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000
	4	0.027	0.000	0.000	**0.948**	0.002	0.000	0.023
	5	0.002	0.000	0.002	0.002	**0.933**	0.000	0.000
	6	0.000	0.000	0.000	0.000	0.000	**1.000**	0.000
	7	0.007	0.000	0.000	0.006	0.000	0.002	**0.985**
III	**1**	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.000	**1.000**	0.000	0.000	0.000	0.000	0.000
	3	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000
	4	0.000	0.000	0.000	**0.992**	0.000	0.000	0.008
	5	0.002	0.000	0.000	0.000	**0.998**	0.000	0.000
	6	0.000	0.000	0.000	0.000	0.000	**1.000**	0.000
	7	0.027	0.000	0.000	0.000	0.000	0.000	**0.973**
IV	**1**	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.000	**1.000**	0.000	0.000	0.000	0.000	0.000
	3	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000
	4	0.000	0.000	0.000	**1.000**	0.000	0.000	0.000
	5	0.000	0.000	0.000	0.000	**1.000**	0.000	0.000
	6	0.000	0.000	0.000	0.000	0.000	**1.000**	0.000
	7	0.000	0.000	0.000	0.000	0.000	0.000	**1.000**

Table 4: Transition percentages produced by the re-classification algorithm w.r.t. RPCL clustering

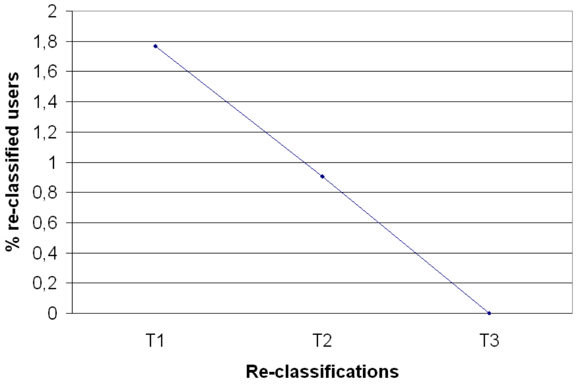

Figure 6: Percentage of re-classified users vs time w.r.t RPCL clustering

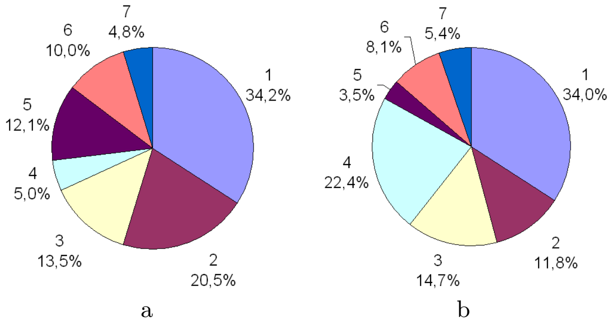

Figure 7: Distribution of users among classes produced a) by RPCL at the time the last re-classification; b) by the last run of the re-classification algorithm

gorithm. Starting from the classification produced by RPCL, the convergence is even faster, due to fact that this classification is better than the one produced by AutoClass C: in fact the value of the C index is lower for the RPCL clustering than for the AutoClass C clustering. Figure 6 shows how the percentage of re-classified users converges towards zero.

Figure 7.a shows the distribution of the users among the classes produced by RPCL on the same day of the last re-classification, while figure 7.b shows the distribution of the users among the classes after the last re-classification.

The last experiment we have carried out consists of evaluating the convergence of the re-classification process as the values of the weight vector W vary. As seen in figure 2, W is used for weighting the initial classification and the contribution of the different resource types. The weight of each type of resource can be defined by an expert of the specific domain, while the weight of the initial classification can be assigned based on how much reliable we consider the registration information provided by the users. In particular, we have set the value of W^r equal to 1.5, 2 and 2.5 when r is equal to *queries*, *directories* and *news/articles*

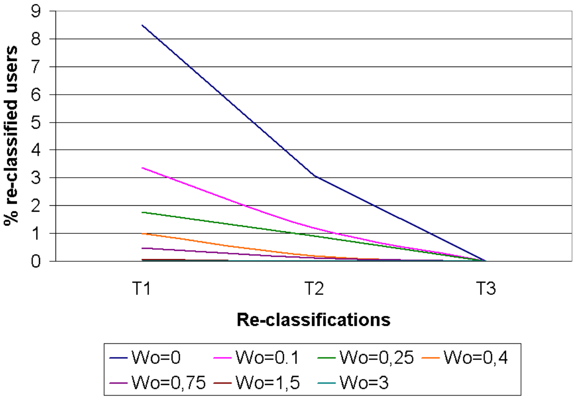

Figure 8: Percentage of re-classified users as W^0 varies

respectively. Varying the value of W^0 from 0 to 3 we have obtained the curves shown in figure 8[2]. We can conclude that the weight assigned to the initial classification can affect the convergence of the re-classification process, making it faster or slower, but in any case a suitable convergence is reached after a few re-classifications.

5. CONCLUSIONS

In this paper we have presented a novel web usage mining strategy for web personalization. The novelty of this strategy relies in the fact that web site users are clustered through a two-phase process that takes into account both static information provided by the users themselves and dynamic behavior. Our approach has been extensively tested on a commercial web site and the experimental results have confirmed the effectiveness of our approach, that leads to a stable classification whatever the initial classification is.

Some aspects needs to be further investigated in the future. In particular we are considering the possibility of designing a re-classification algorithm that, instead of simply moving users from a class to another one, can also dynamically change the number of clusters. This enhancement would be useful to address at least two additional issues, namely: *i*) different clustering algorithms may generate initial classifications with different number of clusters; *ii*) it could be the case that, after some re-classifications, a cluster contains too many or too few elements.

6. REFERENCES

[1] A. G. Büchner and M. D. Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. *ACM SIGMOD Record*, 27(4):54–61, December 1998.

[2] P. Cheeseman and J. Stutz. *Advances in Knowledge Discovery and Data Mining*, chapter Bayesian Classification (AutoClass): theory and results, pages 153–180. 1996.

[3] M. S. Chen, J. S. Park, and P. S. Yu. Data mining for path traversal patterns in a web environment. In *Proceedings of the 16th International Conference on Distributed Computing Systems*, pages 385–392, 1996.

[4] M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, February 2003.

[5] F. Grandi. An annotated bibliography on temporal and evolution aspects in the world wide web. TIMECENTER Technical Report TR-75, University of Bologna, Italy, September 2003.

[6] Z. Huang. A cube model for web access sessions and cluster analysis. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001.

[7] L. Hubert and J. Schultz. Quadratic assignment as a general data-analysis strategy. *British Journal of Mathematical and Statistical Psychologie*, 29:190–241, 1976.

[8] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: a tour guide for the world wide web. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 770–777, August 1997.

[9] H. Lieberman. Letizia: An agent that assists web browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 924–929, 1995.

[10] D. Mladenic. Machine learning used by personal WebWatcher. In *Proceedings of the Workshop on Machine Learning and Intelligent Agents (ACAI-99)*, Chania, Greece, July 1999.

[11] M. Mulvenna, S. S. Anand, and A. G. Buchner. Personalization on the net using web mining. *Communications of the ACM*, 43(8):122–125, August 2000.

[12] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1245–1258, May 1999.

[13] J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan. Web usage mining: Discovery and application of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12–23, January 2000.

[14] K. L. Wu, P. S. Yu, and A. Ballman. Speedtracer: A web usage mining and analysis tool. *IBM Systems Journal*, 37(1), 1998.

[15] L. Xu, A. Krzyzak, and E. Oja. Rival penalized competitive learning for clustering analysis, rbf net and curve detection. *IEEE Transactions on Neural Networks*, 4:636–649, 1993.

[16] T. W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proceeding of the Fifth International World Wide Web Conference*, Paris, 1996.

[17] O. R. Zaiane, M. Xin, and J. Han. Discovering web access patterns and trends by applying olap and data mining technology on web logs. In *Proceedings of Advances in Digital Libraries Conference (ADL98)*, Santa Barbara, CA, April 1998.

[18] F. Zhang and H.Y. Chang. Research and development in web usage mining system-key issues and proposed solutions: a survey. In *Proceedings of the First IEEE International Conference on Machine Learning and Cybernetics*, volume 2, pages 986–990, Beijing, November 2002.

[2] Figure 8 refers to the case in which RPCL is used to perform the initial classification