

# Studying the feasibility of a recommender in a citizen web portal based on user modeling and clustering algorithms\*

José D. Martín-Guerrero (jdmg@uv.es)  
*Digital Signal Processing Group, University of Valencia, Spain*

Alberto Palomares and Emili Balaguer-Ballester  
*R & D Department, TISSAT S.A., Spain*

Emilio Soria-Olivas and Juan Gómez-Sanchis  
*Digital Signal Processing Group, University of Valencia, Spain*

Antonio Soriano-Asensi  
*Applied Physics Department, University of Granada, Spain*

March 29, 2004

**Abstract.** This paper presents a methodology to estimate the future success of a collaborative recommender in a citizen web portal. This methodology consists of four stages, three of them are developed in this study. First of all, a user model, which takes into account some usual characteristics of web data, is developed to produce artificial data sets. These data sets are used to carry out a clustering algorithm comparison in the second stage of our approach. This comparison provides information about the suitability of each algorithm in different scenarios. The benchmarked clustering algorithms are the ones that are most commonly used in the literature: *c-Means*, *Fuzzy c-Means*, a set of *hierarchical algorithms*, *Gaussian mixtures* trained by the *Expectation-Maximization* algorithm, and *Kohonen's Self-Organizing Maps (SOM)*. The most accurate clustering is yielded by SOM. Afterwards, we turn to real data. The users of a citizen web portal (<http://www.infoville.es>) are clustered by using SOM. The clustering achieved enables us to study the future success of a collaborative recommender by means of a prediction strategy. New users are recommended according to the cluster in which they have been classified. The suitability of the recommendation is evaluated by checking whether or not the recommended objects correspond to those actually selected by the user. The results show the relevance of the information provided by clustering algorithms in this web portal, and therefore, the importance of developing a collaborative recommender for this web site.

**Keywords:** recommender, user model, web usage sites, collaborative filtering, citizen web portal.

## 1. Introduction

Recommender systems are a widely used tool in many web sites, such as e-commerce and citizen web portals. The main goal of these systems is

---

\* This work has been partially supported by the Ministerio de Ciencia y Tecnología, Spain, under project FIT-070000-2001-663.



to recommend objects which a user might be interested in (Zukerman & Albretch, 2001). A user model becomes necessary to be able to achieve a useful recommendation. Two main approaches have been used to give recommendations based on user modeling: content-based and collaborative filtering (Zukerman & Albretch, 2001); however, other kinds of techniques have also been proposed (Burke, 2002).

Collaborative recommendation is likely the most widely used technology. Collaborative recommenders aggregate ratings or recommendations of objects, find user similarities based on their ratings, and finally provide new recommendations based on inter-user comparisons. Some of the most important systems using this technique are GroupLens/NetPerceptions (Resnick et al., 1994), Ringo/Firefly (Shardanand & Maes, 1995), and Recommender (Hill et al., 1995). Although some other techniques have been used, the model-based technique is the most widely implemented one for these recommenders. In general, a model is developed from the historical rating data and is used to make predictions (Breese et al., 1998), (Terveen et al., 2002). The greatest strength of collaborative techniques is that they are independent from any machine-readable representation of the objects being recommended, and that they work appropriately for complex objects (for instance, music and movies) where variations in taste are responsible for much of the variation in preferences; this is called “people-to-people correlation” (Schafer et al., 1999).

Content-based learning is used when a user’s past behavior is a reliable indicator of his/her future behavior. Therefore, a predictive model is built for a user by using data from his/her past behavior. Content-based models are particularly suitable for situations in which users tend to exhibit idiosyncratic behavior. However, this approach requires a system to collect relatively large amounts of data from each user in order to enable the formulation of a statistical model. Examples of systems of this kind are text recommendation systems like the news-group filtering system, NewsWeeder (Lang, 1995), which uses words from its texts as features. This kind of learning, where the recommender learns a profile of the user’s interests based on the features present in objects that the user has rated, is called “item-to-item correlation”. As in the case of collaborative filtering, content-based user profiles can be considered as long-term models and can be updated as more evidence about user preferences is observed.

In this paper, we focus on collaborative recommendation since it seems to be a more appropriate technique for citizen web portals since our aim is to find inter-user similarities rather than idiosyncratic behaviors of individual users.

In particular, we propose a four-stage methodology to evaluate a recommender. Other recent works, such as (Geyer-Schulz & Hashler., 2002), also propose similar steps for evaluating a recommender. The main difference between our approach and that presented in (Geyer-Schulz & Hashler., 2002) comes from the third stage of our methodology, which is not considered by the other authors. This stage, which is explained below, is crucial to our approach. Another difference is that we can not develop our fourth stage due to a lack of data, whereas the methodology presented in (Geyer-Schulz & Hashler., 2002) does carry out this last step.

Our methodology starts with a user model which produces artificial data sets, which can be used to compare clustering algorithms, thus finding out the suitability of each algorithm in the different kinds of sets (second stage). This way, once characteristics of real data are known, a clustering according to the most appropriate algorithm can be carried out.

Therefore, the first two stages of our approach involve user modeling and an evaluation of clustering algorithms, respectively. With regard to the clustering algorithms utilized to extract knowledge about user tastes, the familiar *c-Means (CM)*, its fuzzy version (*Fuzzy c-Means (FCM)*), a set of hierarchical clustering algorithms (HCA), and the *Expectation-Maximization (E-M)* algorithm for fitting a Gaussian mixture model were used (Theodoridis & Koutrumbas, 1999). Finally, Kohonen's *Self-Organizing Map (SOM)* (Kohonen, 1997) was also used in this work. Artificial data sets created by the user model were used to benchmark the different clustering algorithms, thus determining the most suitable one for each scenario.

The third and crucial stage of our methodology is related to the evaluation of the feasibility of recommendations with a real data set. Once the users have been clustered, we compare the suitability, or more precisely, the prediction capabilities of a collaborative recommender that utilizes this clustering with a recommender that only recommends the most likely object of the web site that has not yet been accessed. If this "clustering recommender" improves the "trivial recommender" considerably, then we can assume that the former is actually useful in taking into account inter-user similarities for recommendations. These prediction capabilities are called "implicit votes" in (Breese et al., 1998). It is important to point out that we do not measure the influence of the recommendations on the users, which is a phenomenon studied in many recent works (Baudisch & Brueckner, 2002), (Kim, 2002), (Lee, 2002), (McNee, 2003). Instead, we study the capability of the clustering algorithm of profiling user behavior. In fact, our methodology predicts those objects which are accessed by the user without receiv-

ing any recommendations. Therefore, our methodology also enables us to separate the influence of the user interface of the recommendation from the effects of the knowledge extracted by our approach. Yet, once the recommendation system is implemented, it is important to follow up on the success of real recommendations, which will be foreseeably different. In fact, it is logical to think that the success of real recommendations will be better since the presentation of attractive items should affect user behavior positively. The analysis of the effects of real recommendations is the fourth and last stage of the development of a recommender. Most approaches usually skip the third stage, but we think that it is absolutely necessary as a preliminary step in the development of a recommendation engine. It enables us to measure how good the clustering is in terms of profiling user behavior. It can be particularly interesting in certain web portals, in which it is risky to develop a recommender without analyzing its possible effectiveness, because of the expense involved in such development.

The rest of the paper is outlined as follows. Section 2 presents the methodology proposed in this study. Section 3 analyzes the suitability of clustering algorithms as well as our methodology to evaluate the feasibility of a future recommendation system. A discussion about the work is carried out in Section 4, and we present some conclusions and discuss some proposals for further work in Section 5.

## 2. Methods

### 2.1. OUTLINE OF THE APPROACH

Figure 1 shows a general scheme of our approach. Although a more detailed explanation of each part of the scheme is given in the following sections, we begin with a brief description about the general approach.

The first step is one of the most relevant parts of the work: a web-inspired user model. It is used to produce representative artificial data sets, which are used by clustering algorithms to find inter-user similarities. Since different artificial data sets and clustering algorithms are taken into account, we can save the information about the suitability of each algorithm for each kind of data set in a Look-Up Table (LUT). Therefore, when a real data set appears, we can compare its characteristics with the artificial data sets. Once the most similar artificial data set is found, the most suitable algorithm for this set is used to cluster users from the real data set. These tasks include the first two stages of our methodology.

One of the novel contributions of this work is the third stage, which basically consists of a study of the feasibility of a recommender sys-

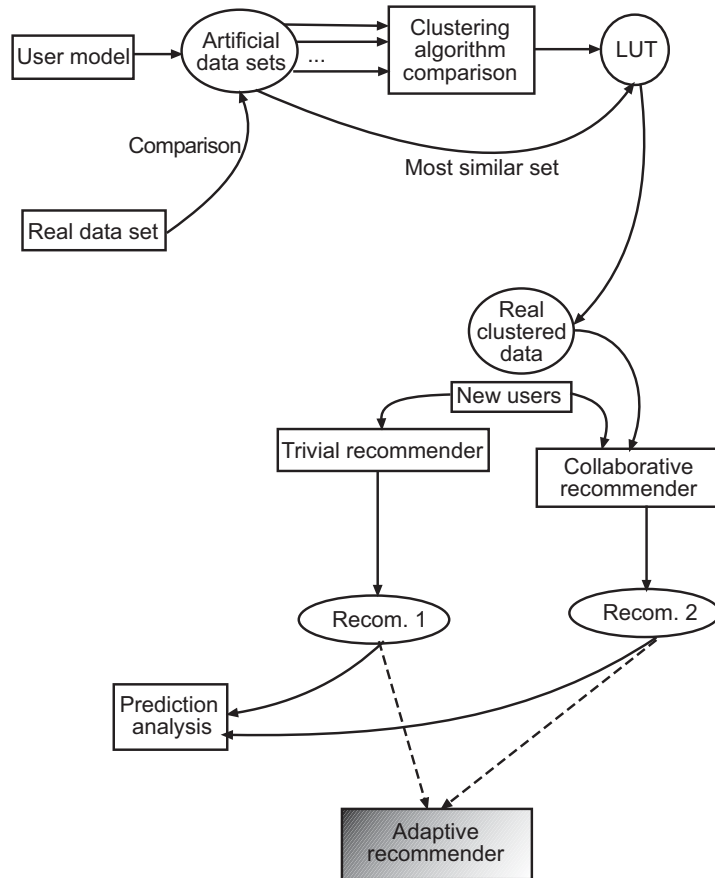
tem. After the clustering of a real data set, new users can be grouped according to this clustering, and thus, they can be recommended by a collaborative strategy. In addition, a trivial system based on recommending only the most likely object that has not yet been accessed is also used in order to compare the benefits of using clustering information. A prediction analysis is performed to determine the suitability of these recommenders, or more particularly, to assess the feasibility of actually implementing these recommenders.

After the actual implementation of these systems, information about the real acceptance of the recommendations by users can be available, and thus, a final study about the actual success of recommendations should be carried out. This task is not included in the present work due to the lack of such data. The recommender could be improved if it included some sort of adaptation in order to fit the recommendations to the users more effectively.

## 2.2. UNDERLYING USER MODEL

Web mining tools must be applicable to real data sets if their practical value is to be realized. However, before turning to real data sets, it is important to characterize their performance; and that is best done initially with artificial data sets for the following reasons:

- *Generalization to web sites with different characteristics.* The application to different web sites is a key point since generic web mining tools should be capable of working properly on different web sites, covering heterogeneous user behaviors. Few real data sets that record user accesses are available because there are more and more restrictive data protection laws and also because of the confidentiality of the web user data kept by the majority of companies. Still, a set might be available, but it would correspond to a particular site, so that if a clustering analysis is carried out on this set, it would only be valid for this site and those sites that have a very similar structure. Nevertheless, artificial data sets can be used to carry out experiments with different site characteristics.
- *Evaluation of algorithm performance.* Before the real application of an algorithm, a rigorous analysis of its performance should be carried out. When dealing with real data, the desired clusters are not usually available *a priori*; hence it is difficult to determine whether the clusters found by the algorithm are right or wrong. However, when an artificial data set is created in a controlled situation, the clusters that must be found by the algorithms are



*Figure 1.* General diagram of our approach. In this diagram, boxes indicate the steps of the methodology, and ellipses indicate the results of a previous step of the user model. The dashed arrows and the gray-shaded square show a future stage of the methodology which has not yet been developed.

defined in advance, thus allowing an analysis of the algorithms' performance.

In this work, we propose a user model, which is a web usage simulator. It creates these artificial data sets. The aim is to develop a user model that is capable of providing a wide range of scenarios. It enables us to do a robust test for the clustering algorithms and, in turn, to determine the most suitable technique for finding user groups in each site. We took into account some of the characteristics and constraints that can be observed in real *log* files (Balaguer & Palomares, 2003), (Breslau et al., 1999), (Andersen et al., 2000), (Su et al., 2000):

- The number of users who log in a new session, i.e, those who access the site, decreases as the number of previously logged-in sessions increases.
- In each session, fewer users access a service (a service is any one of the possible objects that can be clicked on from a web portal) when the number of previously consulted services increases.

These two characteristics are similar to modeling according Zipf’s Law (Breslau et al., 1999). Assuming an exponential decrease<sup>1</sup>, the quantity of users  $N$  that access a certain number of services  $x$  in the  $y^{th}$  session can be obtained from the expression:

$$N = N_M \cdot \exp(-(\alpha \cdot x + \beta \cdot y)) \quad (1)$$

where  $N_M$  is the maximum number of users (those logging in in the first session and accessing at least one object), and  $\alpha$  and  $\beta$  are constants whose values determine the slope of the exponential decrease. Figure 2 shows these restrictions for a particular case generated by the user model. In Figure 3 (a), the percentage of users vs the number of logged-in sessions and (b) the percentage of users vs length of sessions in a real citizen web portal are shown. A strong similarity between the simulated restrictions and the real conditions can be observed.

The user model works in a space of reduced dimensionality because it can be very difficult to find useful inter-user similarities in a space of high dimensionality. Since the quantity of objects that can be clicked on in a web portal may be very large (sometimes, several thousand objects are available), it is not recommended to generate users in a space defined by services; it is preferable to do it in a reduced space instead. It must be taken into account that working with approximately the same or even fewer users than the dimensionality of the space is useless in terms of knowledge discovery. Also, inter-user similarities cannot be found in such a space, either. Therefore, we defined some labels that gather several services with similar characteristics, which led to a lower dimensionality space. These labels are often called “page categories” or “descriptors”; for instance, several pages or objects that are grouped under subject labels like “Football”, “Basketball” and so on (Cadez et al., 2001). However, since descriptors may be unavailable in some cases, the user model offers information about users in a space defined by services as well.

---

<sup>1</sup> Another kind of decrease can be taken into account. For instance, Poisson’s distribution is also quite appropriate to our interests (Yates & Goodman, 1999)

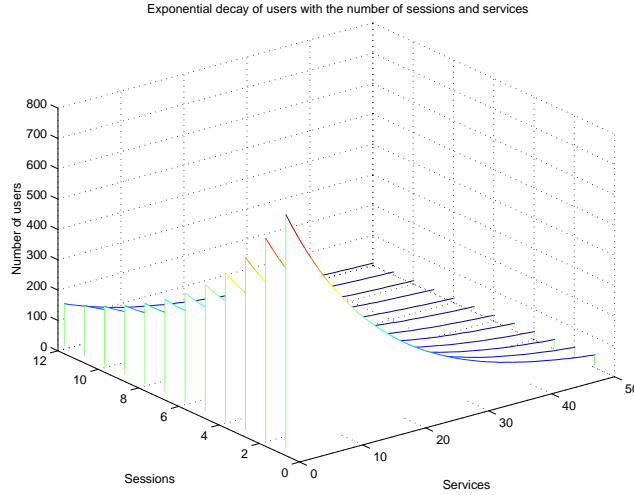


Figure 2. A simulated web site with 50 services and 12 sessions is represented. An exponential decrease of the number of users with respect to the logged-in sessions and the clicked services is shown.

From a global point of view, the system's *modus operandi* consists of two parts, as shown in Figure 4: first, sets of users are generated in a descriptor space, providing a vector for each user. The components of the vector indicate the probability of accessing the descriptors. After this step, the service accesses can be obtained from the relationship between labels and services, and also from the constraints of the user model. Information about label and service accesses is coded into two tensorial matrices. In Figure 4,  $T_D$  is a tensor that records accesses to the different descriptors in each session. Its dimension is  $N \times N_D \times N_{S_{max}}$ , where  $N$  is the number of users,  $N_D$  the number of descriptors and  $N_{S_{max}}$  the maximum number of sessions that can be logged-in by the same user. Let us consider an example to understand the storage of data in  $T_D$ . Assume a portal whose  $N_D = 3$ , and that we want to know the accesses corresponding to user #9 in his/her fourth session. This information is stored in the components  $(9, k, 4)$  of the tensor  $T_D$ , where  $k = 1, 2, \dots, N_D$ . If, for instance,  $T_{D(9,k,4)} = [3, 2, 2]$ , it means that user #9 has logged in to the portal seven times during his/her fourth session, three of which correspond to descriptor  $D_1$ , two to  $D_2$  and the other two to  $D_3$ . Moreover,  $T_S$  is the tensor that records accesses to the different services of the portal in each session. In this case, the dimension of the tensor is  $N \times L_{max} \times N_{S_{max}}$ , where  $L_{max}$  is the maximum length of a session, i.e., the maximum number of services that can be clicked on in only one session. If, analogously to the previous example, we want to know the services accessed by user #9 during his/her fourth

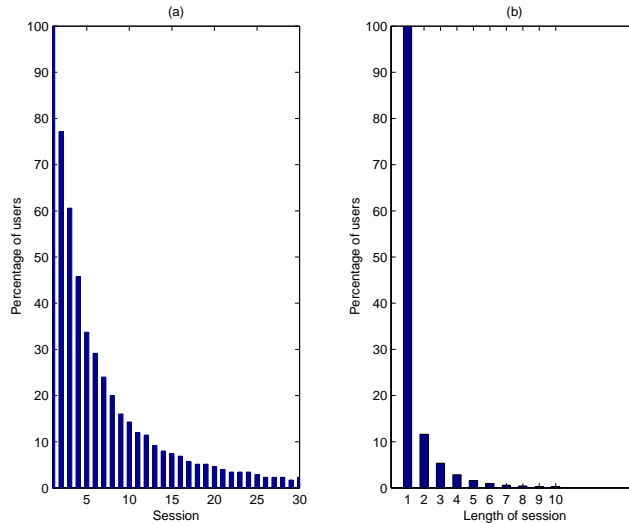


Figure 3. Histograms (normalized to percentages) representing accesses to the citizen web portal *Infoville XXI* (<http://www.infoville.es>). (a) represents the percentage of users vs the number of logged-in sessions; (b) represents the quantity of users vs the length of the session, i.e., the number of clicked services within a session.

session, the result is  $T_{S(9,l,4)} = [43, 27, 2, 6, 22, 19, 5, 0, \dots, 0]$ , where  $l = 1, 2, \dots, L_{max}$ . It means that in his/her fourth session user #9 has clicked on service #43 first, and then on #27, #2, #6, #22, #19 and #5. Therefore, the last selected service is #5, and the user ends his/her navigation in the portal during the fourth session in service #5. The vector is completed with zeros in order to store efficiently the data of users with clickstreams of different lengths.

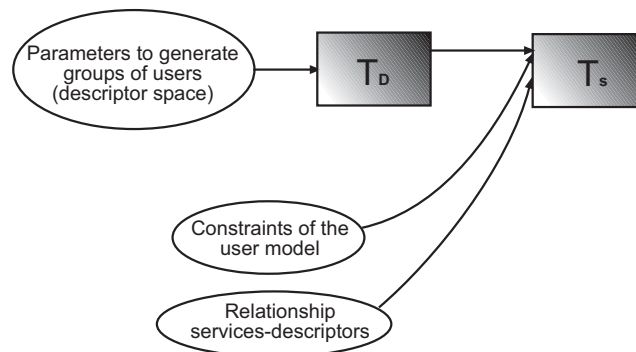


Figure 4. Block diagram showing the stages of the user model.

The parameters of the model that can be chosen are the following: number of services; quantity of descriptors, the “a priori” probabilities

of the descriptors and the services which constitute each descriptor; maximum amount of sessions which could be logged-in by an individual user; length of a session (maximum number of services which a user can click on in one session); number of user groups, users in each group, and statistical parameters, such as the mean and covariance matrix; and finally, parameters controlling the decrease of users vs sessions and services, as shown in (1).

### 2.3. CHARACTERISTICS OF THE ARTIFICIAL DATA SETS

Six artificial data sets were selected in order to test the clustering algorithms. They represent common situations that can occur in web portals since they have been derived from empirical web portal access data (Balaguer & Palomares, 2003), and follow characteristics that are similar to other sets used in the literature (Ghosh et al., 2002), (Banerjee & Ghosh, 2002). The clusters were assumed to follow a normal distribution, so they could be described by the location of their centroids and their covariance or their standard deviation matrix (Gin'e & Zinn., 1986). The artificial data sets were generated in a space defined by the probability of access to descriptors. The main characteristics of each data set are presented, as follows:

#### 2.3.1. *Data set #1*

This is a very simple data set, with just two clusters in a space defined by two descriptors. In contrast to data sets #2-#6, it is not inspired in real-web-portal-access data, but serves purely as a baseline to test the clustering ability in a simple task.

#### 2.3.2. *Data set #2*

This data set is formed by three descriptors, which define the space to cluster, and four groups of users. In this case, the clusters are closer to each other than in data set #1; their shape is not spherical and, in addition, each cluster is formed by a different number of patterns. All these facts make the clusters more difficult to separate. This is shown in Figure 5, where the two data sets are plotted.

The matrices which contain the information about cluster centroids and standard deviations are shown in (2) and (3).

$$C_2 = \begin{pmatrix} 0.1 & 0.25 & 0.5 \\ 0.75 & 0.75 & 0.1 \\ 0.3 & 0.5 & 0.3 \\ 0.5 & 0.1 & 0.9 \end{pmatrix} \quad (2)$$

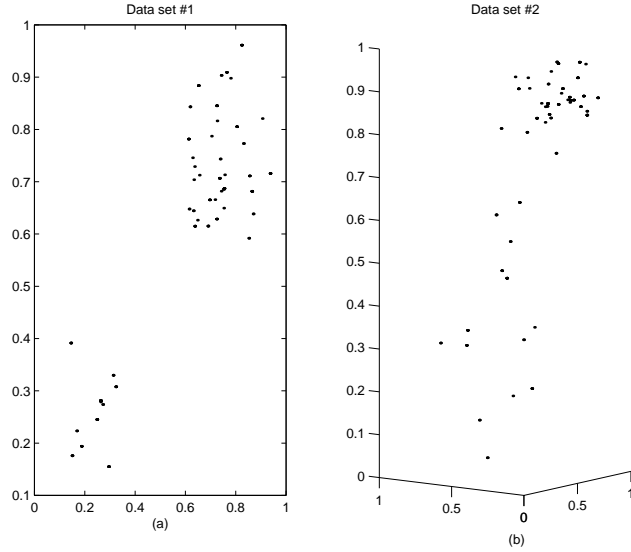


Figure 5. (a) Data set #1: Spherical clusters with a large distance between them; (b) Data set #2: Ellipsoidal shaped clusters with a small distance among them; each cluster is formed by a different number of patterns.

$$\sigma_2 = \begin{pmatrix} 0.1 & 0.08 & 0.04 \\ 0.12 & 0.08 & 0.08 \\ 0.15 & 0.15 & 0.15 \\ 0.2 & 0.1 & 0.05 \end{pmatrix} \quad (3)$$

### 2.3.3. Data sets #3 and #4

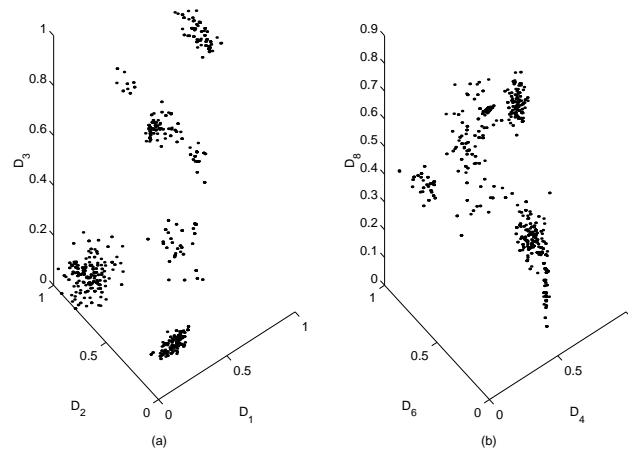
These two data sets consist of eight groups of users in a space of five descriptors. Each cluster is formed by a different number of patterns, as in data set #2. The matrices of centroids and standard deviations were generated pseudo-randomly following a Gaussian distribution for the clusters. Due to the high dimensionality and number of clusters, it is not possible to directly visualize the cluster separation. Standard deviation values are 0.5 in average, and they range from 0.0022 to 0.1118.

With regard to the data set #4, the standard deviation is equal to 0.1 on average and the data range from 0.0031 to 0.2370. Therefore, this set was expected to be more difficult to cluster than data set #3.

### 2.3.4. Data sets #5 and #6

Eight descriptors and twelve groups of users are the principal common characteristics of these two data sets. As in previous data sets, each cluster is formed by a different number of patterns, and the matrices

of centroids and standard deviations were generated pseudo-randomly following a Gaussian distribution for the clusters. The average standard deviation for set #5 was 0.09 (range from 0.0001 to 0.2048), and for set #6 was 0.05 (range from 0.0006 to 0.1142). In Figure 6, two representations of data set #6 are shown.



*Figure 6.* Two different tri-dimensional projections of data set #6 are plotted. In (a), it can be observed that if descriptors  $D_1$ ,  $D_2$  and  $D_3$  are taken into account, some clusters can be distinguished. In (b), nevertheless, if descriptors  $D_4$ ,  $D_6$  and  $D_8$  are taken into account, the observed overlap among clusters is considerable, impeding a useful representation.

#### 2.4. CITIZEN WEB PORTAL DATA: INFOVILLE XXI

Simulated data sets are very useful for carrying out an analysis about algorithm performance in different situations; however, real data become absolutely necessary as a final test. In fact, this is the cornerstone of the methodology proposed; it is based on comparing the real sites with those sites obtained by simulation and, in turn, applying the algorithm that performed the best in the most similar simulated site. In this work, we focus on citizen web portals, an interactive gateway between citizens and the public administration. They involve citizens in the Information Society by offering a growing number of services on the Internet, creating a new model for service delivery to the public as a result of the interaction between the basic services provided by the Government and private entities, which ends up at the citizen who made the request. City or regional portals are the most suitable delivery mechanisms for these services. The success and acceptance of these portals depend largely on

their ability to attract the citizens, and the public and private entities in the area. Finding out inter-user similarities and, in turn, creating groups of users with similar tastes helps in the customization of the portal. This is an easy way to make the site attractive to the majority of the people. In this work, the suitability of customization is analyzed by means of a recommender. This analysis provides information about the possible benefits of carrying out such a customization.

We profiled user accesses to the region web portal *Infoville XXI*, <http://www.infoville.es/>. This is an official web site supported by the Valencian Government, which provides citizens from Valencia, Spain, with more than 2,000 services, grouped into 22 descriptors, namely, public administration, agenda/events, children's area, town councils, street maps, channels<sup>2</sup>, shopping, Infoville community<sup>3</sup>, Infoville diary, education and training, finance, information for citizens, internal, register<sup>4</sup>, Lanetro (local information about where to eat, drink, dance, . . .), SMS messages, entertainment, electronic newspapers, tourism in Valencia, national and international tourism, search and user utilities<sup>5</sup>. More than 50,000 homes are currently connected to Infoville XXI, recording more than 2 million accesses to date. Furthermore, the term *Infoville*, which was coined by the Generalitat Valenciana<sup>6</sup>, is currently part of a European project. In fact, this term is used for citizen web portals from Germany, Italy, England, Denmark and France.

The hierarchical structure of the portal offers the information, first by descriptors, and then, by services. The most popular objects are highlighted, and they can be accessed by clicking on them from the main page. This portal is only available in the two local languages (Valencian and Spanish). Infoville portals from other countries also offer an English version.

We have used accesses from June 2002 to February 2003. The data recorded consists of user ID (a random number which does not offer any information about the identity of the user), session ID and service ID, together with the date and time corresponding to each access. A preprocessing procedure was carried out to eliminate data which did not provide useful information for our goals, and also to build sets

---

<sup>2</sup> Currently, this descriptor consists of information on four specific matters: education, job-hunting, setting up business and housing.

<sup>3</sup> This descriptor enables the communication of people who access the portal by e-mail, fora, postcards, bulletin boards, etc.

<sup>4</sup> Internal and register are descriptors used for administration purposes.

<sup>5</sup> Personal agenda, site customization, personal web page, helping guide, . . .

<sup>6</sup> Generalitat Valenciana is the official name of the autonomous Government of the Valencian Region, in Spain.

for clustering and analysis of the recommendations. This preprocessing procedure involved the following steps:

1. *Removing administrators.* The administrators of the portal create a great number of fictitious users for test purposes. These users are useless in terms of knowledge discovery and, therefore, they were eliminated from the data set. In particular, users who logged in more than 500 sessions were removed.
2. *Removing anomalous users.* Those users who accessed the site only once in all the months included in the study can be considered as lost users. Besides, more than 95% of these users logged in fewer than 30 sessions. In addition, the sum of the mean value and the standard deviation of the data was close to 30 sessions. Therefore, those users who accessed the portal more than 30 times were removed from the data set.
3. *Removing high and low accessed descriptors.* Since the clustering is carried out in the descriptor space, it is important to analyze the information provided by the descriptors. Those descriptors that record a very low number of accesses should be removed because they do not contain an important amount of information. Descriptors that record a very high number of accesses should also be removed, since they can bias the clustering considerably. After this preprocessing procedure, 6 descriptors were eliminated, with 16 descriptors remaining in the data set. This preprocessing step is similar to model descriptors according to Zipf's Law (Breslau et al., 1999). It must be emphasized that these descriptors were removed for clustering tasks, but the services that belonged to them were all taken into account for recommendation.
4. *Removing users who logged in fewer than three times.* Those users who logged in less than three times were removed from the data set, since it would be difficult for the clustering algorithms to find similarities among users with so little information. The final number of users after the preprocessing procedure, was 4,800 users.
5. *Final preparation for clustering.* Accesses to the descriptor "Internal" were not taken into account, since they all correspond to administrators of the portal, and are therefore not useful in terms of recommendation for regular users. Accesses were encoded in a probability notation in order to be processed by the clustering algorithms. Furthermore, data was split into two sets: a first set was used to carry out the clustering (it consisted of 17,404 accesses corresponding to the first half of the months taken into account)

and a second set was used to analyze recommendations (14,079 accesses corresponding to the second half). This analysis is based on whether a recommendation based on the clustering achieved would match the actual services accessed by users. It must be emphasized that this second data set was not used at all for clustering purposes, hence, it enabled us to carry out a recommendation evaluation, and, in turn, to show the robustness of the clustering achieved.

## 2.5. CLUSTERING ALGORITHMS

Five well-known clustering algorithms have been used in order to find inter-user similarities. First, we applied these algorithms to the six simulated data sets, thus discovering the suitability of each algorithm depending on the characteristics of the site. Afterwards, their application to real data from *Infoville XXI* was carried out.

The familiar c-Means (CM) (Theodoridis & Koutrumbas, 1999) was the first algorithm to be taken into account. It can be viewed as a special case of the generalized hard clustering algorithmic scheme when point representatives are used and the *squared* Euclidean distance is adopted to measure the distance between vectors  $x_i$  (users, in our case) and cluster representatives. The algorithm recovers clusters that are as compact as possible.

The fuzzy c-Means (FCM) is the fuzzy version of the c-Means algorithm (Theodoridis & Koutrumbas, 1999). When using this algorithm, users do not belong exclusively to one cluster, but they have a degree of membership degree in different clusters; hence, a user can belong to two or more clusters at the same time with different membership values.

Hierarchical clustering algorithms (HCAs) produce a hierarchy of clusterings. A clustering  $\mathfrak{R}_1$  containing  $k$  clusters is said to be *nested* in the clustering  $\mathfrak{R}_2$ , which contains  $r (< k)$  clusters, if *each* cluster in  $\mathfrak{R}_1$  is a subset of a set in  $\mathfrak{R}_2$  and at least one cluster of  $\mathfrak{R}_1$  is a proper subset of  $\mathfrak{R}_2$ . In this case, we write  $\mathfrak{R}_1 \subset \mathfrak{R}_2$ . HCAs produce a *hierarchy of nested clusterings* (Theodoridis & Koutrumbas, 1999). More specifically, these algorithms involve  $N$  steps, as many as the number of data vectors (users, in our case). At each step  $t$ , a new clustering is obtained based on the clustering produced at the previous step  $t - 1$ . There are two main categories of these algorithms, the *agglomerative* and the *divisive hierarchical algorithms*. In this paper, we focused on the agglomerative ones, since both kinds of algorithms have been widely compared, and agglomerative algorithms have shown better characteristics than divisive ones, mainly in terms of computational burden. The initial clustering  $\mathfrak{R}_0$  for the agglomerative algorithms consists of  $N$  clusters, each

containing a single user. The clustering  $\mathfrak{R}_1$  is produced at the first step. It contains  $N-1$  sets, such that  $\mathfrak{R}_0 \subset \mathfrak{R}_1$ . This procedure continues until the final clustering,  $\mathfrak{R}_{N-1}$ , is obtained, which contains a single set, that is, all the users. Some variants can be considered depending on the way the clusters are joined from one step to the following one. The simplest and most frequently used variants are the single link algorithm and the complete link algorithm. The single link algorithm has a tendency to favour elongated clusters (*chaining effect*), and the complete link algorithm has a tendency to recover small compact clusters.

The Expectation-Maximization algorithm (E-M) maximizes the expectation of the loglikelihood function, which is conditioned on the observed samples, and the current iteration estimate of a parameter vector  $\Phi$ , which initially is unknown (Theodoridis & Koutrumbas, 1999). We used this algorithm for estimation of Gaussian mixtures, where the parameters to estimate are the mean  $\mu$  and the covariance matrix  $\Sigma$ . In this work, we initialized the cluster centers using the CM algorithm in order to have useful groups from the very beginning and in turn, to accelerate the algorithm's convergence.

SOM is an Artificial Neural Network (ANN) used for clustering (Haykin, 1999). It consists of a set of network weights, which are autonomously adapted according to the distribution of input data in the input space (Kohonen, 1997). SOM defines neighborhood relations in the output neurons (neurons that are connected to a neuron with one branch are the nearest to that neuron). When an input is presented to the network, the neuron whose weight vector is the nearest in the input space is selected (it is called, the *firing neuron*). The weight vector of the firing neuron is modified so that it comes closer to the input vector. In addition, the weight vectors of the neurons that are near the firing neuron are modified so that the weight vectors come closer to the input vector. Inside the neighborhood considered, the weight vectors of those neurons that are closer to the firing neuron are moved closer to the input vector than those units which are further from it. The unsupervised learning method that selects one neuron from the competing neurons is called *competitive learning* or *winner-take-all*. In the trained network, neurons are assigned to the points where many data are gathered. Since weights of neurons that are near each other in the neighborhood relations are trained to be near each other in the input space, the network is considered to be a mapping that preserves neighborhood relations in the input space. If the Euclidean distance is used, the adaptation of the weight vectors  $\mathbf{w}_j$  of the neurons near the firing neuron (including the weight vector of the firing neuron) is given

by (4):

$$\mathbf{w}_j^{new} = \mathbf{w}_j^{old} + \varepsilon h_{ij} (\mathbf{x} - \mathbf{w}_j^{old}) \quad (4)$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_{N_D})^t$  is the  $N_D$ -dimensional input vector ( $N_D$  is the number of descriptors),  $\varepsilon$  is a positive constant that satisfies  $0 < \varepsilon < 1$  and  $h_{ij}$  is a neighborhood function determined by the distance between neurons  $i$  and  $j$ ,  $i$  being the winner neuron. The neighborhood function  $h_{ij}$  may be given by the step function in which  $h_{ij}$  is constant within some specified distance, and the value of the function is zero outside the specified distance or the Gaussian function:

$$h_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \quad (5)$$

where  $d_{ij}$  is a distance determined by the neighborhood relation between neurons  $i$  and  $j$ , and  $\sigma^2$  is the variance of the distribution which can be adjusted. Parameters  $\varepsilon$  and  $\sigma$  are decreased during training to prevent the weight vectors from oscillating. If we associate each training datum with the firing neuron after training, then the weight vector of the firing neuron is the representative or prototype value of the training data. Since the weight vectors of the neighboring neurons are corrected as well as that of the firing neuron, the data compaction is performed while retaining the neighborhood relation of the training data. To use the SOM for clustering, the network is trained by setting the number of neurons to more than the number of classes and training the network using all the training data several times. The result is a map (usually, a two-dimensional map), which shows the neighborhood relations among data. This map is a clustering representation. In this work, a digital image processing stage was carried out to segment different clusters in the map representation more clearly. After training, we input a class datum into the SOM and associated each datum with the firing neuron. The set of data associated with a neuron forms a cluster; this procedure enables us to know the cluster prototype in the original space, i.e., in the space defined by the probability of descriptors.

We used a procedure that is based on joining close neurons, which tended to favour elongated clusters. Due to this fact, we used a two-stage methodology. First, we carried out a first clustering by using SOM (we searched for a number of clusters higher than that actually expected), and afterwards, we used a complete link HCA to find the final clusters, which tended to favour compact clusters, thus correcting the preference for elongated clusters of the first stage.

### 3. Results

#### 3.1. ALGORITHM COMPARISON IN ARTIFICIAL DATA SETS

As we explained in Section 2.3, six artificial data sets were selected for algorithm comparison. They represented different kinds of web usage sites and were produced by the user model already explained. Since these data sets were simulated, we knew the clusters that should be found; hence, we could evaluate the performance of the different algorithms. In order to evaluate the clusters achieved, two measures were taken into account. On the one hand, we considered whether the number of clusters found by the algorithm was correct or not, and, on the other hand, how good these clusters turned out to be.

Therefore, we compared the number of prototypes found by the algorithms with the number we knew in advance. Of course, we measured absolute values for the difference between the number of clusters found and the desired one (finding more or fewer classes than the actual ones is considered as an error). The final Success Rate (SR [%]) of correct clusters found by the different algorithms is compared in Table I. This SR is relative to the actual number of groups, and is calculated as follows:

$$SR[\%] = 100 \cdot \left( \frac{N_a - |N_a - N_f|}{N_a} \right) \quad (6)$$

where  $N_a$  is the actual number of groups and  $N_f$  the number of correct clusters found by the algorithm.

The goodness of the clustering was measured by the Mahalanobis distance from each cluster found by the algorithm<sup>7</sup> to the nearest known cluster' center. The advantage of using this distance is that it takes into account the covariance of the group, hence it does not depend on the shape of the cluster. We took into account the number of patterns in each cluster to evaluate the goodness of the clustering:

$$D = \frac{1}{N} \sum_{i=1}^{N_f} N_i d_i \quad (7)$$

In (7),  $D$  provides information about the distance from the found clusters to the actual centers. The smaller the value of  $D$ , the closer the match to the known clusters.  $N$  is the whole number of patterns,  $N_i$  is the number of patterns belonging to the  $i$ -th cluster found by the algorithm, and  $d_i$  is the Mahalanobis distance from the  $i$ -th found cluster to

---

<sup>7</sup> When the number of patterns which forms a class is very small, this distance is not useful, and the Euclidean one is used instead.

the actual center.  $D$  values for the different algorithms and data sets are benchmarked in Table II. This table must be analyzed in combination with Table I. The best overall behavior was shown by SOM, since only E-M and HCA algorithms showed a better performance in one data set (specifically, set #4). It is important to point out that CM and FCM presented a lower value of  $D$  in this data set, but their SR was worse. In fact, CM only presented an acceptable behavior in set #1, which is quite simple. This suggests that this widely used algorithm is not appropriate to cluster data sets of this kind. Besides this, FCM showed a slightly better behavior, and, E-M and HCA, in turn, had a better behavior. As a general conclusion, when the data set to be clustered is supposed to be difficult, or the dimensionality is high or the characteristics are not known “a priori”, then SOM should be the algorithm used.

Table I. Success Rate (SR) [%] of correct clusters found by the algorithms with respect to the actual number of groups.

	CM	FCM	HCA	E-M	SOM
Set #1	100	100	100	100	100
Set #2	25.0	25.0	50.0	75.0	50.0
Set #3	37.5	62.5	75.0	50.0	75.0
Set #4	37.5	37.5	75.0	75.0	75.0
Set #5	30.0	41.7	66.7	58.3	66.7
Set #6	0	41.7	58.3	66.7	58.3

### 3.2. ROBUSTNESS OF THE CLUSTERING USING ARTIFICIAL DATA SETS

A final test of the algorithms’ robustness was carried out by analyzing the normality of the clusters achieved, given that the artificial data sets were generated with multivariate Gaussian distributions. For this purpose, measures of *skewness* and *kurtosis*, can be used (Hair et al., 1998). Skewness is a measure of symmetry, or more precisely, the lack of symmetry:

$$skewness = \frac{\sum_{i=1}^N (x_i - \bar{X})^3}{(N - 1)\sigma^3} \quad (8)$$

Table II. Normalized Mahalanobis distance (D) between actual centers and the correct clusters found by the different algorithms. The distances are measured in the space defined by probability of access to descriptors.

	CM	FCM	HCA	E-M	SOM
Set #1	0.0330	0.0330	0.1125	0.0330	0.0165
Set #2	0.6818	0.0262	0.0682	0.9097	0.0819
Set #3	0.1498	0.7846	0.1880	0.3091	0.0797
Set #4	0.2227	0.2051	0.2600	0.2861	0.3101
Set #5	0.2858	0.3583	0.2615	0.2738	0.2403
Set #6	...	0.2134	0.2016	0.5892	0.0418

where,  $x_i$  is the  $i$ -th data point,  $\bar{X}$  is the mean (prototype of the cluster),  $\sigma$  the standard deviation, and  $N$  is the number of data points. The skewness for a normal distribution is zero, and any symmetric data should have a skewness near zero. Negative values for the skewness indicate data that are skewed to the left and positive values for the skewness indicate data that are skewed to the right.

On the other hand, kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. That is, data sets with high kurtosis tend to have a distinct peak near the mean, decline rather rapidly and have heavy tails. Data sets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. A uniform distribution would be the extreme case:

$$kurtosis = \frac{\sum_{i=1}^N (x_i - \bar{X})^4}{(N-1)\sigma^4} \quad (9)$$

The kurtosis for a standard normal distribution is three. For this reason, excess kurtosis is defined as:

$$kurtosis_{exc} = \frac{\sum_{i=1}^N (x_i - \bar{X})^4}{(N-1)\sigma^4} - 3 \quad (10)$$

so that the standard normal distribution has an excess kurtosis of zero. A positive kurtosis indicates a peaked distribution and a negative kurtosis indicates a flat distribution.

A statistical test based on skewness and kurtosis values can be carried out to assess normality. The statistic value ( $z$ ) for the skewness

and kurtosis values are calculated as:

$$z_{skewness} = \frac{skewness}{\sqrt{\frac{6}{N}}} \quad (11)$$

$$z_{kurtosis} = \frac{kurtosis_{exc}}{\sqrt{\frac{24}{N}}} \quad (12)$$

If the calculated  $z$  value exceeds a critical value, then the distribution is non-normal in terms of that characteristic (Hair et al., 1998). The critical value is from a  $z$  distribution, based on the significance level we desire. For example, a calculated value exceeding  $\pm 2.58$  indicates we can reject the assumption about the normality of the distribution at the .01 error level. Another commonly used critical value is  $\pm 1.96$ , which corresponds to a .05 error level.

The statistic value  $z$  for the skewness and kurtosis values was analyzed for the distributions found by the clustering algorithms with the simulated data sets. Values near zero were desired since data were simulated following a normally distributed random sequence. We used the critical value  $\pm 1.96$ ; if a higher value was obtained, we considered the distribution as non-normal. The most important result found was that, for those sets in which SOM, E-M, and HCA showed similar results in terms of goodness of clustering, HCA presented a high percentage of non-normal clusters, sometimes nearly 50%; even E-M, which assumes a Gaussian mixture presented higher percentages of non-normal clusters than SOM. Nevertheless, SOM clustering was very normal, and  $z$  values were within the margin of critical values considered.

In summary, having analyzed the behavior of algorithms in several simulated usage sites, the conclusion is that if the site is a very simple one, the CM algorithm is enough; thus, it is not worth using more complex algorithms. When more complex sites are presented, the results achieved suggest that the choice should be SOM, although E-M and HCA also presented acceptable results. However, the use of SOM is encouraged by the authors since the results achieved regarding non-normality tests suggest that it has greater capabilities to grasp the statistical characteristics of the cluster.

### 3.3. PRELIMINARY CLUSTERING OF REAL DATA: INFOVILLE XXI

First, a preliminary study was carried out just to know the capabilities of the algorithms to find useful and understable clusters for this citizen web portal. This issue was assessed by selecting a small group from the available descriptors. A reduced data set (November 2002 through January 2003) was used. First, the access frequencies of each descriptor

were analyzed to remove the least discriminant descriptors. From the remaining descriptors, five were selected by Tissat, S.A.<sup>8</sup> as the most significant ones: public administration, town councils, channels, shopping and entertainment. This resulted in 1,676 users for this study. We used 1,000 users to obtain the clusters, and we kept the other 676 to test the robustness of the clustering achieved. The users of each group were selected randomly.

The results were analyzed in terms of the interpretability of the clusters obtained. This was possible because the clustering was done in a five-dimension space, in which the meaning of all the components was known. A comparison among algorithms showed that the clustering achieved by FCM, E-M and CM was not easy to understand. In fact, the clusters did not represent logical behaviors of people. However, the clustering achieved by HCA, and SOM were straightforward, since they clustered the data into seven different groups: five of them were clearly focused on each one of the five different descriptors, whereas the other two clusters contained people who were interested in the leisure items of the portal or in the administrative ones. In particular, one of the clusters was centered between the descriptors “shopping” and “entertainment”. Therefore, it clustered individuals who mainly accessed the portal for leisure purposes. The other cluster was centered between the descriptors “public administration” and “town councils”, and it also presented a small membership to the descriptor “channels”. Therefore, people clustered in this group clearly accessed the portal for administrative purposes. These seven clusters demonstrate two important facts: on the one hand, HCA and SOM seem to be suitable as clustering tools for this portal; on the other hand, the usefulness of the portal is clearly demonstrated, since it was basically designed to accomplish these two requirements, i.e., to accelerate administrative paperwork, and to provide a fast gateway for the leisure interests of citizens.

Therefore, in order to decide which algorithm should be used for the final clustering (the one that takes into account all the descriptors), FCM, E-M and CM were rejected, since if they did not work appropriately in a simple version of the portal, they will probably not work with the real, more complex version. If one has to make a decision between HCA and SOM, the choice should be SOM. Although both these algorithms present a similar behavior with the reduced version of the portal, the SOM showed the best performance with high-dimensional artificial data sets, which suggests that it will also present the best behavior in high-dimensional, real data sets. Moreover, the SOM used

---

<sup>8</sup> Tissat, S.A. is the company responsible for developing the portal.

in this study is actually formed by a SOM and a HCA that are chained, as mentioned above.

### 3.4. REAL DATA: FINAL CLUSTERING AND FEASIBILITY ANALYSIS OF A RECOMMENDER

Finally, the data set formed by the users of *Infoville XXI* described in Section 2.4 was clustered using SOM. Since the clusters were not known in advance, the evaluation described in Section 3.1 could not be carried out, nor was it feasible to analyze the interpretability of the groups obtained due to the high-dimensional space in which the clustering was performed. The evaluation of the clustering can be assessed by means of analyzing the success of the recommendations based on this clustering. This is a new approach which can be used not only to evaluate the clustering, but also, and mainly, to study the feasibility of a recommender before its actual implementation.

Clustering is used to carry out a kind of collaborative filtering, i.e., the most likely service of the user group is recommended, provided that this service has not yet been accessed. Services based on clustering cannot be recommended for the first accesses, since there is not enough information to assign users to a certain cluster. Instead, the most likely services of the portal are recommended for these first accesses, providing that they have not yet been accessed.

Afterwards, a test is performed to determine whether or not users actually click on the recommended object. Finally, the Success Ratio (SR) achieved in the prediction of the accessed services by using our methodology is benchmarked with that SR obtained by recommending only the most likely service of the whole portal that has not yet been accessed. Therefore, the effectiveness of our methodology is measured in terms of the improvement in the SR with respect to the methodology which does not use clustering.

Different mappings yielded by SOM were analyzed<sup>9</sup>. A clustering formed by eight groups of users, and produced by a two-dimensional SOM was chosen. A feasibility analysis was carried out in a service domain, although the clustering was done in a space defined by the probability of access to descriptors. This analysis involved a two-step process. First, for the  $m$  first accesses of a user, the most probable service of the whole portal not previously accessed by this user is considered for recommendation. Second, in the  $n$ -th access to the portal ( $n \geq m+1$ ), the previous  $n-1$  accesses are used to measure the distance

---

<sup>9</sup> Davies-Bouldin and Dunn indexes were used in order to choose the most appropriate number of clusters (Bezdek, 1998).

between user behavior and the eight different groups, selecting the one which shows the minimal distance as the *winner* cluster.

After this, the most likely descriptor within the winner cluster is chosen, and then, the most likely service belonging to this descriptor (provided that it has not yet been accessed) is considered for recommendation. We consider a success to be when the object considered for recommendation is actually clicked on. In order to analyze the improvement of using clustering information, we consider different values of  $m$ . We also consider different values  $l$  of the number of accesses for which we analyze the prediction ( $l \geq m + 1$ ).

The Average Success Ratio (ASR) over the 14,076 accesses used for the evaluation is benchmarked in Table III for different values of  $m$  and  $l$ . It can be observed that our methodology based on using clustering information yields higher ASRs than the methodology that does not take into account this information. As more accesses are used to cluster, better results are obtained; this is expected, since the information gathered by the clustering algorithms is more extensive. Besides this, the importance of the clustering appears to be more relevant in the first accesses starting from the  $(m + 1)$ -th one; as the number of accesses increase, the difference between using clustering information or not becomes smaller. Therefore, clustering appears to be particularly important in the first accesses of the users, when they must be attracted in order to establish their loyalty to the portal.

#### 4. Discussion

Recommender systems are one of the most prolific fields of research and publication of user modeling. In this work, we focus our efforts on recommendation systems for web sites, although their application to other areas is also possible with some small changes. A good recommender is undoubtedly useful since users can achieve the objects searched for in less time, or even better, find something interesting that they would not have found by themselves. It is also useful for the company which exploits the site, since obvious economical profits can be obtained from useful recommendations. Finally, a good recommender also provides an indirect benefit, which is the improvement of the web site.

However, the development of such systems is not easy, and in addition, it may involve a high economic investment. Until now, recommender systems have been developed and then evaluated; in this

Table III. Average Success Rate (ASR) [%] measuring the goodness of service prediction as a preliminary step in the development of a recommender. Prediction with and without clustering is benchmarked for different values of  $m$  and  $l$ .

<b>m</b>	<b>l</b>	<b>No Clustering</b>	<b>Clustering</b>
<b>2</b>	<b>4</b>	6.91	12.84
<b>2</b>	<b>5</b>	10.12	14.57
<b>2</b>	<b>6</b>	13.07	16.48
<b>2</b>	<b>7</b>	16.13	18.74
<b>3</b>	<b>4</b>	3.47	13.73
<b>3</b>	<b>5</b>	7.04	15.11
<b>3</b>	<b>6</b>	10.31	16.81
<b>3</b>	<b>7</b>	13.70	18.97
<b>4</b>	<b>6</b>	7.56	18.06
<b>4</b>	<b>7</b>	11.32	19.94
<b>5</b>	<b>7</b>	8.16	20.94

work, we propose a novel approach, which consists of carrying out an evaluation of the recommender before being implemented in order to analyze its feasibility. It is based on predicting the objects that will be rated by the users. If this prediction works, then the users have been successfully profiled. Thus, it is logical to believe that a recommender system using a similar strategy would work even better, since attractive recommendations can affect user behavior, making the users click on such recommendations. Therefore, the success obtained with the prediction could be interpreted as a lower threshold of the success that can be obtained with a similar recommender system.

In particular, we have benchmarked a prediction based on using information about user clustering with the one that predicts the most likely service of the portal that has not yet been accessed by the recommended user. More specifically, we use clustering of users to classify new users in a certain group, thus finding out which descriptor will be the most likely for this user, and afterwards, predicting the most probable service of this descriptor not yet accessed by the user. In order to make the prediction useful from a recommendation point of view, it is important not to predict/recommend services already accessed by users, since these objects are already known by them, and therefore,

they do not provide new information about the portal, which is one of the most important goals of a recommender system.

The results show that using clustering information provides a much better prediction, showing success rates which are approximately double the rates obtained with respect to prediction without this information. Although it might seem obvious, the authors want to point out that the users used for clustering are different from the users used to evaluate the prediction. This demonstrates the robustness of the clustering achieved, and the relevance of the information provided by it.

In order to choose the clustering tool, several well-known clustering algorithms have been benchmarked in some artificial data sets, which have been obtained by a user model; the data sets represent different kinds of web usage sites. Moreover, these algorithms have also been benchmarked by using a reduced data set of the web portal *Infoville XXI*, and by analyzing the interpretability of the clustering achieved. All these tests show that SOM is the most suitable technique of all the techniques compared in this work.

Part of the approach proposed in this work is already implemented in the software iSUM<sup>®</sup> (<http://www.isum.com/>); our aim is to include all the methodology in future versions.

## 5. Conclusions and future work

A novel approach to evaluate the feasibility of implementing a recommender in web portals is proposed in this work. The first step of the methodology is to cluster users by using a SOM. It serves to classify new users who log in to the portal. Once they are classified, the services that will be accessed by them are predicted, according to a useful recommender scheme, i.e., the prediction offers the most likely services that will be accessed, provided that they have not yet been accessed.

The results show a considerable improvement with respect to a prediction which does not take into account clustering information. This suggests the implementation of a recommender following this strategy in the web portal analyzed in this study.

In spite of the fact that we have focused on a citizen web portal in this work, this methodology can also be applied to other web portals, and even to other fields, for instance, interactive TV. The latter would imply a change in the user model, which is currently designed for web usage sites.

Future work will be dedicated to carrying out a follow-up of real recommendations once these data are available. This follow-up should

be used to improve the recommender, since feedback of actual recommendations' can be used to adapt the system. Many techniques can be utilized; schemes based on Learning Vector Quantization appear to be a promising tool to achieve this goal.

### Acknowledgements

The authors would like to express their thanks to *Fundació OVSI (Oficina Valenciana per a la Societat de la Informació—Valencian Office for Information Society)* for their permission to analyze their data and for their interest in the conclusions of this study. We would also like to thank *Tissat, S.A.* for their collaboration and technical support.

### References

- Abe, S. *Pattern classification: neuro-fuzzy methods and their comparison*. Springer Verlag, London, 2001.
- Alpaydin, E. Soft vector quantization and the E-M algorithm. *Neural Networks*, 11:467–477, 1998.
- Andersen, J., Larsen, R. S, Giversen, A., Pedersen, T. B, Jensen, A. H. and Skyt, J. Analyzing Clickstreams Using Subsessions. Technical Report TR-00-5001, Department of Computer Science, Aalborg University, 2000.
- Balaguer, E. and Palomares, A. AI Recommendation Engine of TISSAT, S.A.. Internal Technical Report, TISSAT, S.A., 2003.
- Banerjee, A., and Ghosh, J. Characterizing visitors to a Web site across multiple sessions. In *Proceedings of NGDM02: National Science Foundation Workshop on Next Generation Data Mining*, Marriott Inner Harbor, Baltimore, MD, USA, 2002.
- Baudisch, P., and Brueckner, L. TV scout: lowering the entry barrier to personalized TV program recommendation. In P. De Bra, R. Conejo, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, Proceedings of the 2nd International Conference, AH2002*, Málaga, Spain, pp. 58–68, 2002.
- Bezdek, J. C., and Pal, N. R. Some New Indexes of Cluster Validity. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):301–315, 1998.
- Breslau, L., Cao, P., Fan, L., Phillips, G. and Shenker, S. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proceedings of INFOCOM 1999*, 1:126–134, 1999.
- Breese, J. S., Keckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998.
- Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User Adapted Interaction*, 12:331–370, 2002.
- Cadez, I., Heckerman, D., Meek, C., Smyth, P. and White, S. Model-Based Clustering and Visualization and Navigation Patterns on a Web Site. Technical Report MSR-TR-0018, Microsoft Research, Microsoft Corporation, 2001.

- Geyer-Schulz, A., and Hashler, M. Evaluation of Recommender Algorithms for an Internet Information based on Simple Association Rules and on the Repeat-Buying Theory. In *Proceedings of WEBKDD02*, Edmonton, Canada, pp. 100–114, 2002.
- Gin'e, E., and Zinn, J. *Lectures on the central limit theorem for empirical processes*. Lect. Notes in Math. 1221. Springer-Verlag, New York, NY, USA, pp. 50–112, 1986.
- Ghosh, J., Strehl, A., and Meregu, S. A Consensus Framework for Integrating Distributed Clusterings Under Limited Knowledge Sharing. In *Proceedings of NGDM02: National Science Foundation Workshop on Next Generation Data Mining*, Marriott Inner Harbor, Baltimore, MD, USA, pp. 99–108, 2002.
- Haykin, S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, USA, 1999.
- Hair, J. F., Anderson, R. E., Tatham R. L. and Black, W. C. *Multivariate data analysis*. Prentice Hall, Upper Saddle River, NJ, USA, fifth edition, 1998.
- Hill, W., Stead, L., Rosenstein, M. and Furnas, G. Recommending and Evaluating choices in a virtual community of use. In *CHI'95: Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, pp. 194–201, 1995.
- Kim, Y., Ok, S. and Woo, Y. A case-based recommender system using implicit rating techniques. In P. De Bra, R. Conejo, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, Proceedings of the 2nd International Conference, AH2002*, Málaga, Spain, pp. 522–526, 2002.
- Kohonen, T. *Self-organizing Maps*. Springer Verlag, Berlin, second edition, 1997.
- Lang, K. Newsweeder: Learning to filter news. In *Proceedings of the 12th International Conference on Machine Learning*, Lake Tahoe, CA, pp. 331–339, 1995.
- Lee, M., Choi, P. and Woo, Y. A hybrid recommender system combining collaborative filtering with neural networks. In P. De Bra, R. Conejo, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, Proceedings of the 2nd International Conference, AH2002*, Málaga, Spain, pp. 531–534, 2002.
- McNee, S. M., Lam, S. K., Konstan, J. A. and Riedl, J. Interfaces for eliciting new user preferences in recommender systems. In P. Brusilowsky, A. Corbett, F. de Rosis, editors, *User Modeling 2003, Proceedings of the 9th International Conference*, Johnstown, PA, pp. 178–187, 2003.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, pp. 175–186, 1994.
- Schafer, J. B., Konstan, J. and Riedl, J. Recommender Systems in E-Commerce. In *EC'99 Proceedings of the First ACM Conference on Electronic Commerce*, Denver, CO, pp. 158–166, 1999.
- Shardanand, U. and Maes, P. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *CHI'95: Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, pp. 210–217, 1995.
- Su, Z., Ye-Lu, Q. Y and Zhang, H. J. WhatNext: A Prediction System for Web Requests using N-gram Sequence Models. In *WISE 2000 Proceedings: 1st International Conference on Web Information Systems Engineering*, Hong Kong, pp. 214–221, 2000.
- Terveen, L. G., McMackin, J., Amento, B., and Hill, W. Specifying Preferences Based On User History. In *CHI 2002: Proceedings of CHI 2002*, Minneapolis MN, pp.315–322.

- Theodoridis, S. and Koutroumbas, K. *Pattern Recognition*. Academic Press, 1999.
- Yates, R. D. and Goodman, D. J. *Probability and Stochastic Processes*. John Wiley & Sons, 1999.
- Zukerman, I. and Albrecht, D. W. Predictive Statistical Models for User Modeling. *User Modeling and User Adapted Interaction*, 11:5–18, 2001.

