



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)



Expert Systems  
with Applications

Expert Systems with Applications xxx (2006) xxx–xxx

[www.elsevier.com/locate/eswa](http://www.elsevier.com/locate/eswa)

# An approach based on the Adaptive Resonance Theory for analysing the viability of recommender systems in a citizen Web portal

José D. Martín-Guerrero <sup>a,\*</sup>, Paulo J.G. Lisboa <sup>b</sup>, Emilio Soria-Olivas <sup>a</sup>,  
Alberto Palomares <sup>c</sup>, Emili Balaguer <sup>c</sup>

<sup>a</sup> Digital Signal Processing Group, Electronics Engineering Department, University of Valencia, Spain

<sup>b</sup> The Statistics and Neural Computation Research Group, School of Computing and Mathematical Sciences,  
Liverpool John Moores University, United Kingdom

<sup>c</sup> Tissat, S.A., R&D Department, Spain

## 12 Abstract

13 This paper proposes a methodology to optimise the future accuracy of a collaborative recommender application in a citizen Web portal. There are four stages namely, user modelling, benchmarking of clustering algorithms, prediction analysis and recommendation. The first stage is to develop analytical models of common characteristics of Web-user data. These artificial data sets are then used to evaluate the performance of clustering algorithms, in particular benchmarking the ART2 neural network with K-means clustering. Afterwards, it is evaluated the predictive accuracy of the clusters applied to a real-world data set derived from access logs to the citizen Web portal *Infoville XXI* (<http://www.infoville.es>). The results favour ART2 algorithms for cluster-based collaborative filtering on this Web portal. Finally, a recommender based on ART2 is developed. The follow-up of real recommendations will allow to improve recommendations by including new behaviours that are observed when users interact with the recommender system.

21 © 2006 Elsevier Ltd. All rights reserved.

22 *Keywords:* Adaptive Resonance Theory; User profiling; Web citizen portal; Recommender systems

## 24 1. Introduction

25 Web mining has become an important research area from the 90s. This is because the huge popularity of the Web and the wide range of possibilities that it offers. One of the most important research efforts within Web mining is that related with finding interesting characteristics and patterns of the Web users and their usage of the Web. The importance of this kind of Web mining is that if users are correctly profiled, then it is possible to understand their behaviour in the portal, and in turn, to provide suitable services for them (Fu, Shandu, & Shih, 1999), especially where this can successfully anticipate demand by individual users.

36 In fact, the study and development of personalized recommender systems is a very active field of research (Carberry, 2001), and some recommender systems become an important part of some Web sites providing e-commerce services, for instance, Amazon.com (<http://www.amazon.com>) and its subsidiary “CDNow” (<http://www.cdnow.com>). There are two main automatic approaches for recommendations which have been extensively tested and are scale-up to large amount of data, namely collaborative filtering and content-based (Zukerman & Albrecht, 2001) recommender systems.

37 Collaborative filtering is among the most widely used technologies today. These recommender systems aggregate ratings or other indicators of interest for web objects, such as frequency of access, to find user similarities based on indicator profiles and thus finally offer recommendations for new pages, services or products. Well-known recom-

\* Corresponding author. Tel.: +34 963160198; fax: +34 963160466.  
E-mail address: [jose.d.martin@uv.es](mailto:jose.d.martin@uv.es) (J.D. Martín-Guerrero).

52 mender systems include GroupLens/NetPerceptions  
 53 (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994),  
 54 Ringo/Firefly (Shardanand & Maes, 1995), and Recom-  
 55 mender (Hill, Stead, Rosenstein, & Furnas, 1995). The  
 56 greatest strength of collaborative techniques is that they  
 57 are independent from any machine-readable representation  
 58 of the objects being recommended and they work appropri-  
 59 ately for complex objects (for instance, music and movies)  
 60 where variations in taste are responsible for much of the  
 61 variation in preferences, sometimes called “people-to-peo-  
 62 ple correlation” (Schafer, Konstan, & Riedl, 1999).

63 Content-based learning is used when a user’s past  
 64 behaviour is a reliable indicator of his/her future behav-  
 65 iour. Content-based models are particularly suitable for sit-  
 66 uations in which users tend to exhibit idiosyncratic  
 67 behaviour. However, this approach requires a system to  
 68 collect relatively large amounts of data from each user in  
 69 order to enable the formulation of a statistical model. Typ-  
 70 ical examples of systems of this kind are text recommenda-  
 71 tion systems like the newsgroup filtering system,  
 72 NewsWeeder (Lang, 1995) which uses words from its texts  
 73 as features. This kind of learning, where the recommender  
 74 learns a profile of the user’s interests based on the features  
 75 present in objects that the user has rated, is called “item-to-  
 76 item correlation”.

77 In this paper, we focus on people-to-people collabora-  
 78 tive recommendation since it seems to be a more appropri-  
 79 ate technique for citizen Web portals since our aim is to  
 80 find inter-user similarities rather than idiosyncratic behav-  
 81 iours of individual users. In particular, our approach con-  
 82 sists of profiling users’ behaviour by using clustering  
 83 algorithms, thus finding groups of similar users, and after-  
 84 wards, recommending those objects in which the users will  
 85 likely be interested in; this knowledge about users’ tastes is  
 86 extracted from the analysis of the services that are usually  
 87 accessed by the users of the same group. The approach of  
 88 user modelling, – by means of clustering algorithms or  
 89 other techniques, – as a first stage of a collaborative recom-  
 90 mender system is not unusual; an example of this kind of  
 91 systems is Moonranker, a free access recommender of  
 92 music, movies and books (Zhou, Weston, Gretton, &  
 93 Schölkopf, 2003).

94 In particular, we propose a four-stage methodology to  
 95 develop and evaluate a clustering recommender (Martin  
 96 et al., 2006): user model, clustering algorithms’ compar-  
 97 ison, prediction analysis and recommendation (Fig. 1).  
 98 Other recent works, such as (Geyer-Schulz & Hashler,  
 99 2002) also propose similar steps for evaluating a recom-  
 100 mender. The main difference between our approach and  
 101 that presented in Geyer-Schulz and Hashler (2002) comes  
 102 from the third stage of our methodology, which is novel.  
 103 The proposed methodology starts with a user model  
 104 which produces artificial data sets which, in the second  
 105 stage of the methodology, serves to evaluate clustering  
 106 algorithms’ performance, in order to benchmark the pre-  
 107 dictive accuracy of the algorithms for the different data  
 108 sets.

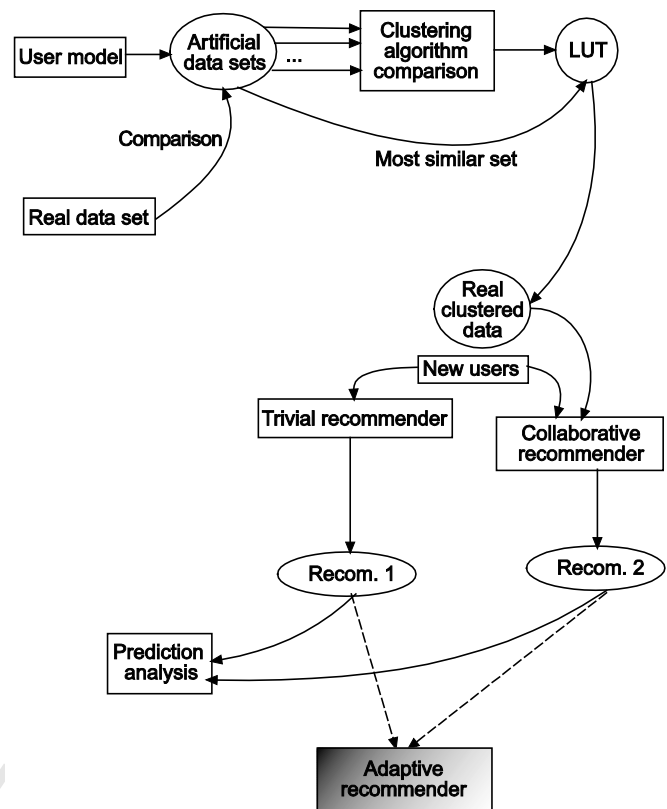


Fig. 1. General schematic of the proposed methodology for the design of cluster-based collaborative filters for web portals. Ellipses indicate the results of a previous step of the user model, and the dashed arrows and the gray-shaded square show a planned extension of the methodology (LUT means look-up table).

The benchmarking of clustering algorithms, focuses on  
 the use of neural networks based on the Adaptive Reso-  
 nance Theory network, ART2 (Carpenter & Grossberg,  
 1987, 1991) which applies to continuous data. This net-  
 work can circumvent some of the usual drawbacks of clas-  
 sical algorithms, having been designed to solve the  
*stability–plasticity* dilemma, namely, the ability to adapt  
 clusters to new data patterns, without disrupting the  
 already established clusters. This feature can support on-  
 line tracking user profiles, although this is not tested in  
 the current study. In practice, this works by identifying  
 the most appropriate cluster for a given user pattern, then  
 testing whether the cluster prototype is a good-enough rep-  
 resentation of the user pattern, which generates clusters  
 with similar distance distributions but of different sizes,  
 whereas the K-means algorithm (Duda, Hart, & Stork,  
 2001) tends to create clusters of similar sizes but potentially  
 with large differences in distance distributions from one  
 cluster to another. The ART2 algorithm is tuned with the  
 desired degree of similarity, or maximum separation  
 among patterns from the same group, rather than by pre-  
 specifying the overall number of clusters in the data. A  
 novel component of the proposed methodology is the use  
 of artificial data created from a sample of user accesses in

133 order to refine and tune the clustering algorithms which, in  
134 this study, demonstrated accuracy improvements gained  
135 using the neural network model.

136 The third and **critical** stage of our methodology is  
137 related to the evaluation of the viability of recommenda-  
138 tions with a real data set. Once the Web users have been  
139 clustered, we compare the suitability, or more precisely,  
140 the prediction accuracy of a collaborative recommender  
141 that utilizes ART2 clustering with another collaborative  
142 recommender based on K-means clustering and with  
143 another recommender that only recommends the most  
144 likely object of the Web site that has not yet been accessed.  
145 If there is a considerable improvement when using a clus-  
146 tering recommender, then we can assume that it is actually  
147 useful taking into account inter-user similarities for recom-  
148 mendations. These prediction capabilities are called “implicit  
149 votes” in Breese, Keckerman, and Kadie (1998). It is  
150 important to point out that we do not measure the influ-  
151 ence of the recommendations on the users, which is a phe-  
152 nomenon studied in many recent works (Baudisch &  
153 Brueckner, 2002; Kim, Ok, & Woo, 2002; Lee, Choi, &  
154 Woo, 2002; McNee, Lam, Konstan, & Riedl, 2003).  
155 Instead, we study the capability of the clustering algorithm  
156 for profiling user behaviour. In fact, our methodology pre-  
157 dicted those objects which are accessed by the user without  
158 receiving any recommendations. Therefore, our methodol-  
159 ogy also enables the influence of the user interface of the  
160 recommendation to be separated from the effects of the  
161 knowledge extracted by our approach. Yet, once the rec-  
162 ommendation system is implemented, it is important to fol-  
163 low up on the success of real recommendations, which will  
164 in general be different. In fact, it is logical to think that the  
165 success of real recommendations will be better than the  
166 success of our prediction analysis. This is because the pre-  
167 sentation of attractive items should affect user behaviour  
168 positively (Cosley, Shyong, Albert, Konstan, & Riedl,  
169 2003; McNee et al., 2003).

170 Most approaches usually skip the third stage, but we  
171 think that it is absolutely necessary as a preliminary step  
172 in the development of a recommender system. It enables  
173 us to measure how good the clustering is in terms of profil-  
174 ing user behaviour. It can be particularly interesting in  
175 certain Web portals, in which it is risky to develop a recom-  
176 mender without analysing its possible effectiveness, because  
177 of the expense involved in such development. The analysis  
178 of the effects of real recommendations is the fourth and last  
179 stage of the development of a recommender system.

180 The remainder of the paper is outlined as follows.  
181 Section 2 presents the data sets used in this study. Sec-  
182 tion 3 analyses our proposal for clustering recommenda-  
183 tion. Section 4 shows the results achieved in this study,  
184 analysing the clustering achieved with the different data  
185 sets as well as our study to evaluate the feasibility of a  
186 future recommendation system. A discussion about the  
187 work is carried out in Section 5, and we present some  
188 conclusions and discuss some proposals for further work  
189 in Section 6.

## 2. Data sets 190

### 2.1. Artificial data sets 191

#### 2.1.1. A user model of Web accesses 192

193 Web mining tools must be applicable to real data sets.  
194 However, the use of artificial data sets also becomes very  
195 important because of the following reasons: 195

- *Artificial data sets enable us to choose the most appropri- 196*  
*ate clustering method with real data.* This is a major rea- 197  
son for the use of artificial data sets. Before the real 198  
application of an algorithm, a rigorous analysis of its 199  
performance should be carried out. When dealing with 200  
real data, the desired clusters are not usually available 201  
a priori; hence it is difficult to determine whether the 202  
clusters found by the algorithm are right or wrong. 203  
However, when an artificial data set is created in a con- 204  
trolled situation, the clusters that must be found by the 205  
algorithms are defined in advance, thus allowing an 206  
analysis of the algorithms' performance. 207
- *Generalization to Web sites with different characteristics.* 208  
Web mining tools should be capable of working prop- 209  
erly on different Web sites, covering heterogeneous user 210  
behaviours. Few real data sets that record user accesses 211  
are available because there are more and more restric- 212  
tive data protection laws and also because of the confi- 213  
dentiality of the Web user data kept by the majority 214  
of companies. Still, a set might be available, but it 215  
would correspond to a particular site, so that if a clus- 216  
tering analysis is carried out on this set, it would only 217  
be valid for this site and those sites that have a very 218  
similar structure. However, artificial data sets can be 219  
used to carry out experiments with different site 220  
characteristics. 221

222  
223 In this work, artificial data sets are generated by a Web  
224 user model which is capable of providing a wide range of  
225 scenarios. This user model takes into account some of the  
226 characteristics and constraints that can be observed in real  
227 log files (Andersen et al., 2000; Balaguer & Palomares,  
2003; Breslau, Cao, Fan, Phillips, & Shenker, 1999; Su,  
Ye-Lu, & Zhang, 2000), namely: 228  
229

- The number of users who log in a new session, i.e., those 230  
who access the site, decreases as the number of previ- 231  
ously logged-in sessions increases. 232
- In each session, fewer users access a service (a service is 233  
any one of the possible objects that can be clicked on 234  
from a Web portal) when the number of previously 235  
consulted services increases. 236

237 These two characteristics are similar to modelling accord- 237  
ing Zipf's Law (Breslau et al., 1999). Assuming an expo- 238  
nential decrease (Fig. 2), the quantity of users  $N$  that 239  
access a certain number of services  $x$  in the  $y$ th session 240  
can be obtained from the expression: 241

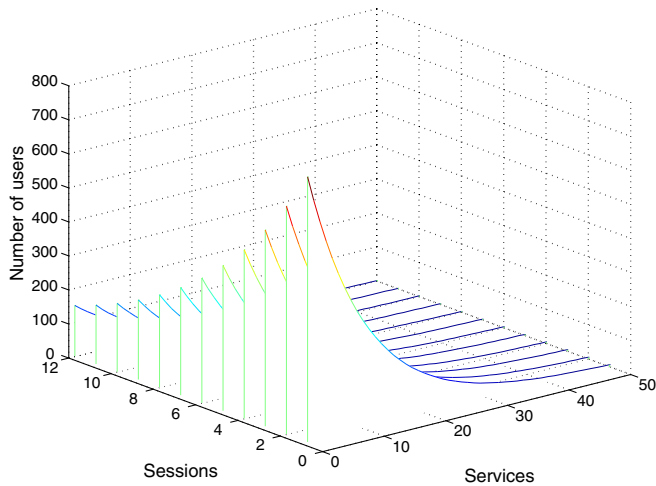


Fig. 2. A simulated Web site with 50 services and 12 sessions is represented. An exponential decrease of the number of users with respect to the logged-in sessions and the clicked services is shown.

243 
$$N = N_M \cdot e^{-(\alpha \cdot (x-1) + \beta \cdot (y-1))}$$
 (1)

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

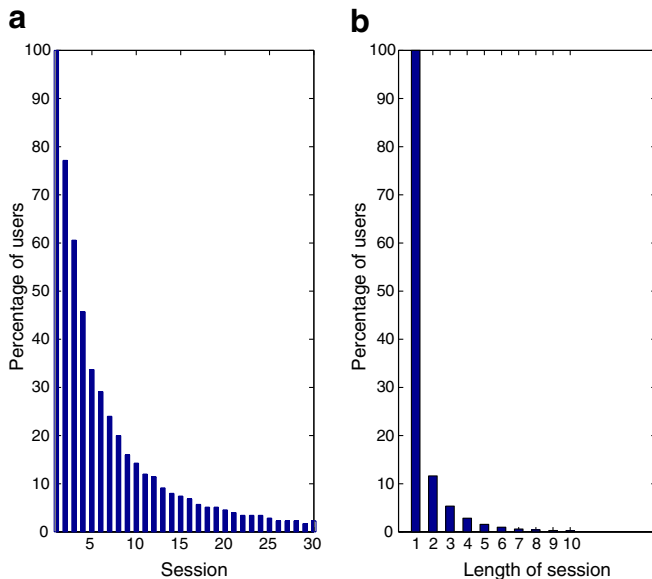


Fig. 3. Histograms (normalized to percentages) representing accesses to the citizen Web portal *Infoville XXI* (<http://www.infoville.es>). (a) Represents the percentage of users vs the number of logged-in sessions; (b) Represents the percentage of users vs the length of the session, i.e., the number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

254 The user model works in a space of reduced dimensionality because it can be very difficult to find useful inter-user  
 255 similarities in a space of high dimensionality. Since the  
 256 quantity of objects that can be clicked on in a Web portal  
 257 may be very large, it is not recommended to generate users  
 258 in a space defined by services; it is preferable to do it in a  
 259 reduced space instead. It must be taken into account that  
 260 working with approximately the same or even fewer users  
 261 than the dimensionality of the space is useless in terms of  
 262 knowledge discovery. Also, inter-user similarities cannot  
 263 be found in such a space, either. Therefore, we defined  
 264 some labels that gather several services with similar char-  
 265 acteristics, which led to a lower dimensionality space. These  
 266 labels are often called “page categories” or “descriptors”;  
 267 for instance, in an electronic newspaper, one can consider  
 268 several pages or objects that are grouped under subject  
 269 labels like “Sport”, “Politics” and so on (Cadez, Hecker-  
 270 man, Meek, Smyth, & White, 2001). However, since  
 271 descriptors may be unavailable in some cases, the user  
 272 model offers information about users in a space defined  
 273 by services as well.  
 274

275 The user model consists of two main parts, as shown in  
 276 Fig. 4: first, sets of users are generated in a descriptor  
 277 space, providing a vector for each user. The components  
 278 of these vectors indicate the a priori probability of access-  
 279 ing the descriptors. After this step, the service accesses can  
 280 be obtained from the relationship between labels and ser-  
 281 vices, and also from the constraints of the user model.  
 282 Information about label and service accesses is coded into  
 283 two tensorial matrices. In Fig. 4,  $T_D$  is a tensor that records  
 284 accesses to the different descriptors in each session. Its  
 285 dimension is  $N \times N_D \times N_{Smax}$ , where  $N$  is the number of  
 286 users,  $N_D$  the number of descriptors and  $N_{Smax}$  the maxi-  
 287 mum number of sessions that can be logged-in by the same  
 288 user. Let us consider an example to understand the storage  
 289 of data in  $T_D$ . Assume a portal whose  $N_D = 3$ , and that we  
 290 want to know the accesses corresponding to user #9 in his/  
 291 her fourth session. This information is stored in the compo-  
 292 nents  $(9, k, 4)$  of the tensor  $T_D$ , where  $k = 1, 2, \dots, N_D$ . If,  
 293 for instance,  $T_{D(9,k,4)} = [3, 2, 2]$ , it means that user #9 has  
 294 accessed seven objects during his/her fourth session, three  
 295 of which correspond to descriptor  $D_1$ , two to  $D_2$  and the

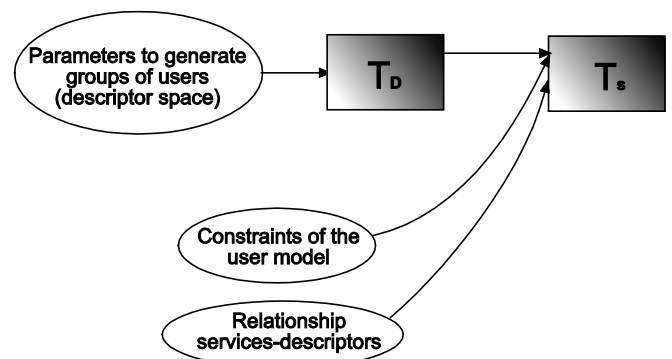


Fig. 4. Block diagram showing the stages of the user model.

296 other two to  $D_3$ . Moreover,  $T_S$  is the tensor that records  
 297 accesses to the different services of the portal in each  
 298 session. In this case, the dimension of the tensor is  
 299  $N \times L_{\max} \times N_{S_{\max}}$ , where  $L_{\max}$  is the maximum length of  
 300 a session, i.e., the maximum number of services that can  
 301 be clicked on in only one session. If, analogously to the  
 302 previous example, we want to know the services accessed  
 303 by user #9 during his/her fourth session, the result might  
 304 be  $T_{S(9,l,4)} = [43, 27, 2, 6, 22, 19, 5, 0, \dots, 0]$ , where  $l =$   
 305  $1, 2, \dots, L_{\max}$ . It means that in his/her fourth session user  
 306 #9 has clicked on service #43 first, and then on #27, #2,  
 307 #6, #22, #19 and #5. Therefore, the last selected service  
 308 is #5, and the user ends his/her navigation in the portal  
 309 during the fourth session in service #5. The vector is com-  
 310 pleted with zeros in order to store efficiently the data of  
 311 users with click-streams of different lengths.

### 312 2.1.2. Characteristics of the artificial data sets

313 Six artificial data sets were selected in order to test the  
 314 clustering algorithms. They represent common situations  
 315 that can occur in Web portals since they have been derived  
 316 from empirical Web portal access data (Balaguer & Palo-  
 317 mares, 2003), and follow characteristics that are similar  
 318 to other sets used in the literature (Banerjee & Ghosh,  
 319 2002; Ghosh, Strehl, & Meregu, 2002). The clusters were  
 320 assumed to follow a normal distribution, so they could  
 321 be described by the location of their centroids and their  
 322 covariance or their standard deviation matrix. The artificial  
 323 data sets were generated in a space defined by the probabil-  
 324 ity of access to descriptors. The main characteristics of  
 325 each data set are presented in Table 1.

326 Data set #1 is a very simple data set, with just two clusters  
 327 in a space defined by two descriptors. In contrast to the other  
 328 data sets, it is not inspired in real-Web-portal-access data,  
 329 but serves purely as a baseline to test the clustering ability  
 330 in a simple task. Data set #2 is considerably more complex,  
 331 especially because the clusters are very close to each other,  
 332 showing a high overlap. Data sets #3 and #4 are similar  
 333 since they consist of eight groups of users in a space of five  
 334 descriptors; the difference between them stems from the  
 335 overlap, which is higher in the case of data set #4. Finally,  
 336 data sets #5 and #6 represent accesses to Web portals in a  
 337 high dimension, since eight descriptors are taken into

Table 1  
Main characteristics of the artificial data sets

	$N_D$	$N_C$	Overlap
Data set #1	2	2	No
Data set #2	3	4	High
Data set #3	5	8	Slight
Data set #4	5	8	High
Data set #5	8	12	High
Data set #6	8	12	Slight

$N_D$  represents the number of descriptors and  $N_C$  the number of clusters. Moreover, the degree of overlap among the different clusters is also shown (we consider a slight overlap when less than 20% of the patterns are overlapped, whereas a high overlap means that more than 20% of the patterns are overlapped among different clusters).

account. Data sets with a higher number of descriptors were  
 also created. However, clustering algorithms showed very  
 similar results to those obtained with data sets #5 and #6.  
 Therefore, they were not selected for benchmarking of clus-  
 tering algorithms. In other words, data sets #5 and #6 are a  
 good-enough representation of high-dimensional data sets.

## 2.2. A real data set: accesses to the Web portal Infoville XXI

### 2.2.1. Characteristics of the data set

In spite of simulated data sets are very useful for carry-  
 ing out an analysis about algorithm's performance in differ-  
 ent situations, real data become absolutely necessary as a  
 final test. 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 ser-  
 vices provided by the Government and private entities,  
 which ends up at the citizen who made the request. The  
 success and acceptance of these portals depend largely on  
 their ability to attract the citizens, and the public and pri-  
 vate 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 peo-  
 ple. In this work, the suitability of customization is anal-  
 ysed 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 *Info-  
 ville XXI*, <http://www.infoville.es>. This is an official Web  
 site supported by the Valencian Government, which pro-  
 vides citizens from Valencia, Spain, with more than 2000  
 services, grouped into 22 descriptors, namely, public  
 administration, agenda/events, children's area, town coun-  
 cils, street maps, channels (this descriptor consists of infor-  
 mation on four specific matters: education, job-hunting,  
 setting up a business and housing), shopping, Infoville  
 community (this descriptor enables the communication of  
 people who access the portal by e-mail, fora, postcards,  
 bulletin boards, etc.), Infoville diary, education and train-  
 ing, finance, information for citizens, internal, register  
 (internal and register are descriptors used for administra-  
 tion purposes), Lanetro (local information about where  
 to eat, drink, dance, ...), SMS messages, entertainment,  
 electronic newspapers, tourism in Valencia, national and  
 international tourism, searcher and user utilities (personal  
 agenda, site customization, personal Web page, helping  
 guide, ...). Furthermore, the term *Infoville*, which was  
 coined by the Generalitat Valenciana,<sup>1</sup> is currently part

<sup>1</sup> Generalitat Valenciana is the name of the autonomous government of Valencia.

389 of an European project. In fact, this term is used for citizen  
390 Web portals from Germany, Italy, England, Denmark and  
391 France.

### 392 2.2.2. Preprocessing

393 We have used accesses from June 2002 to February  
394 2003. The data recorded consists of user ID, session ID  
395 and service ID, together with the date and time corre-  
396 sponding to each access. A preprocessing procedure was  
397 carried out to eliminate data which did not provide useful  
398 information for our goals, and also to build sets for cluster-  
399 ing and analysis of the recommendations. This preprocess-  
400 ing procedure involved the following steps:

- 401 • *Removing administrators.* The administrators of the por-  
402 tal create a great number of fictitious users for test pur-  
403 poses. These users are useless in terms of knowledge  
404 discovery and, therefore, they were eliminated from  
405 the data set.
- 406 • *Removing anomalous users.* Those users who accessed  
407 the site only once in all the months included in the study  
408 can be considered as lost users, and therefore, they were  
409 removed from the data set. Besides, more than 95% of  
410 the users logged in fewer than 30 sessions, being  
411 removed those users who accessed the portal more than  
412 30 times.
- 413 • *Removing high and low accessed descriptors.* Since the  
414 clustering is carried out in the descriptor space, it is  
415 important to analyse the information provided by the  
416 descriptors. Those descriptors that record a very low  
417 number of accesses should be removed because they  
418 do not contain an important amount of information.  
419 Descriptors that record a very high number of accesses  
420 should also be removed, since they can bias the cluster-  
421 ing considerably. After this preprocessing procedure, six  
422 descriptors were eliminated, with 16 descriptors remain-  
423 ing in the data set. It must be emphasized that these  
424 descriptors were removed for clustering tasks, but the  
425 services that belonged to them were all taken into  
426 account for recommendation.
- 427 • *Removing users who logged in fewer than three times.*  
428 Those users who logged in less than three times were  
429 removed from the data set, since it would be difficult  
430 for the clustering algorithms to find similarities among  
431 users with so little information. The final number  
432 of users after the preprocessing procedure, was 4800  
433 users.
- 434 • *Final preparation for clustering.* Accesses were encoded  
435 in a probability notation in order to be processed by  
436 the clustering algorithms. Furthermore, data was split  
437 into two sets: a first set was used to carry out the cluster-  
438 ing (it consisted of 17,404 accesses corresponding to  
439 the first half of the months taken into account) and a  
440 second set was used to analyse recommendations  
441 (14,079 accesses corresponding to the second half of  
442 the period of time taken into account). The latter anal-  
443 yses whether a recommendation based on the clustering

achieved would match the actual services accessed by 444  
users. It must be emphasized that this second data set 445  
was not used at all for clustering purposes, hence, it 446  
enabled us to carry out a recommendation evaluation, 447  
and, in turn, to show the robustness of the clustering 448  
achieved. 449  
450

## 3. Recommendations based on clustering 451

### 3.1. Clustering with ART 452

The ART model was originally proposed by Carpenter 453  
and Grossberg (1987) to model fast adaptive learning in 454  
the initial stages of human visual processing. Hence it is 455  
termed an artificial neural network. In its initial form, 456  
ART1, the model applied only to clustering of binary vec- 457  
tors. It remains among few clustering methods specifically 458  
designed for quantized data. The model was then extended 459  
to continuous-valued vectors in ART2 (Carpenter & 460  
Grossberg, 1991). These networks cluster inputs by using 461  
unsupervised learning. 462

ART operates as a two stage process. Each time a pat- 463  
tern is presented, an appropriate cluster unit is chosen, 464  
and that cluster's weights are adjusted to let the cluster unit 465  
learn the pattern. The weights on a cluster unit are consid- 466  
ered to be a prototype for the patterns assigned to that 467  
cluster. The second and crucial stage of the recognition 468  
process is to test whether the prototype forms and adequate 469  
representation of the input pattern. Once a good-enough 470  
winning prototype has been selected, the process is referred 471  
to a vigilance test. From this, either the prototype is 472  
updated to form a running average of the input vector, 473  
or a new prototype is initiated. 474

As a computational tool, ART networks allow the user 475  
to control the degree of similarity of patterns placed on the 476  
same cluster; once this choice is done, it is not necessary to 477  
choose the number of clusters in advance, but the network 478  
finds the number corresponding to the degree of similarity 479  
chosen. During training, each data pattern is presented sev- 480  
eral times. A pattern may be placed on one cluster unit the 481  
first time it is presented and then placed on a different cluster 482  
when it is presented later (due to changes in the weights 483  
for the first cluster if it has learned other patterns in the 484  
meantime). A stable network will not return a pattern to 485  
a previous cluster, i.e., a pattern oscillating among different 486  
cluster units at different stages of training indicates an 487  
unstable network. Some self-organized neural network 488  
models achieve stability by gradually reducing the learning 489  
rate as the same set of training patterns is presented many 490  
times (Kohonen, 1997). However, this does not enable the 491  
network to learn rapidly a new pattern that is presented for 492  
the first time after a number of training epochs have 493  
already taken place. The ability of a network to respond 494  
to a new pattern equally well at any stage of learning is 495  
called *plasticity*. ART networks are designed to be both 496  
stable and plastic. 497

498 In this work, we used ART2 network<sup>2</sup> first to cluster  
 499 patterns from the artificial data sets presented earlier. Since  
 500 artificial data sets enable us to analyse clustering perfor-  
 501 mance, we benchmark the clustering achieved by ART2  
 502 with that obtained by using the classical K-means. As it  
 503 is shown later, ART provide much better results, thus  
 504 showing its capabilities to cluster this kind of data sets;  
 505 ~~moreover, ART was also benchmarked with a wide range~~  
 506 ~~of clustering methods in a previous work (Martin, 2003),~~  
 507 ~~being a very good method to cluster these data sets.~~ After-  
 508 wards, ART2 was also applied to cluster users from the  
 509 Web portal *Infoville XXI*; since in this case we are working  
 510 with real data, evaluation of the clustering is carried out by  
 511 studying the meaning of the clusters found and also analy-  
 512 sing the success of recommendations based on clustering  
 513 (prediction analysis).

### 514 3.2. Procedure of recommendations

515 Clustering of users of the citizen Web portal *Infoville*  
 516 *XXI* was used to carry out a kind of collaborative filtering,  
 517 i.e., the most likely service of the user group was recom-  
 518 mended, provided that this service had not yet been  
 519 accessed. Services based on clustering could not be recom-  
 520 mended for the first accesses, since there was not enough  
 521 information to assign users to a certain cluster. Instead,  
 522 the most likely services of the portal were recommended  
 523 for these first accesses, providing that they had not yet been  
 524 accessed.

525 After the recommendations, a test was performed to  
 526 determine whether or not users actually clicked on the rec-  
 527 ommended object; since ART clustering was much more  
 528 accurate than K-means clustering in all scenarios, recom-  
 529 mendations should be based on ART. The success ratio  
 530 (SR) achieved in the prediction of the accessed services  
 531 by using an ART clustering was benchmarked with that  
 532 SR obtained by recommending only the most likely service  
 533 of the whole portal that had not yet been accessed. There-  
 534 fore, the effectiveness of our methodology was measured in  
 535 terms of the improvement in the SR with respect to a meth-  
 536 odology which did not use clustering.

537 Although the clustering is carried out in a space defined  
 538 by the probability of accesses to descriptors, the analysis of  
 539 viability of developing a recommender was carried out in a  
 540 service domain. This analysis involved a two-step process.  
 541 First, for the  $m$  first accesses of a user, the most probable  
 542 service of the whole portal not previously accessed by this  
 543 user was considered for recommendation. Second, in the  
 544  $n$ th access to the portal ( $n \geq m + 1$ ), the previous  $n - 1$   
 545 accesses were used to measure the distance between user's  
 546 behaviour and the clusters found by the algorithm, select-  
 547 ing the one which shows the minimum distance as the *win-*  
 548 *ner* cluster. After this, the most likely service within the  
 549 winner cluster is chosen (provided that it has not yet been

accessed), and considered for recommendation. We con- 550  
 sider a success to be when the object considered for recom- 551  
 mendation is actually clicked on. We consider different 552  
 values of  $m$ , and also different values of  $l$ , being  $l$  the num- 553  
 ber of accesses for which we analyse the prediction 554  
 ( $l \geq m + 1$ ). 555

## 556 4. Results

### 557 4.1. Clustering of artificial data sets

558 In order to evaluate the clusters achieved, two 558  
 approaches were taken into account. On the one hand, we 559  
 considered whether or not the number of clusters found 560  
 by the algorithm was correct, and, on the other, how good 561  
 these clusters were. 562

563 Therefore, we compared the number of prototypes 563  
 found by the algorithms with the correct number that we 564  
 knew in advance, and afterwards, the goodness of the clus- 565  
 tering was measured by the Mahalanobis distance from 566  
 each cluster found by the algorithm to the nearest known 567  
 cluster' centre.<sup>3</sup> The advantage of using this distance mea- 568  
 sure is that it takes into account the covariance of the 569  
 group, hence it does not depend on the shape of the cluster. 570  
 Any cluster whose Mahalanobis distance from the nearest 571  
 known cluster centre was  $>1$ , did not match the corre- 572  
 sponding centre properly, and was removed from the set 573  
 of correct groups. Finally, we took account of the number 574  
 of patterns in each cluster in the final measure to evaluate 575  
 the quality of the clustering: 576

$$577 D = \frac{1}{N} \sum_{i=1}^M N_i d_i \quad (2) \quad 579$$

580 In Eq. (2),  $D$  provides information about the distance from 580  
 the cluster found to the nearest actual centre. The smaller 581  
 the value of  $D$ , the closer the match to the known cluster. 582  
 $N$  is the whole number of patterns,  $M$  the number of cor- 583  
 rect clusters found,  $N_i$  the number of patterns belonging 584  
 to the  $i$ th cluster found, and  $d_i$  the Mahalanobis distance 585  
 from the  $i$ th cluster found to the corresponding centre. 586

587 The number of clusters found by K-means and ART2 587  
 are compared in Fig. 5. When the dimensionality is low, 588  
 similar results were achieved by both algorithms, but 589  
 ART2's behaviour was much better when dealing with a 590  
 high number of clusters in a high dimensionality space. It 591  
 is important to emphasize that K-means had information 592  
 about the number of clusters in advance, which ART2 593  
 did not have. Nevertheless, ART2 achieved better results 594  
 than K-means clustering. This seems to be an advantage 595  
 of the two-stage similarity used by ART2, which success- 596  
 fully filters similarities among data, the final number of 597  
 clusters being a natural result of these similarities; on the 598  
 other hand, K-means tries to find a certain number of 599

<sup>2</sup> A detailed procedure of the algorithm is shown in Appendix A.

<sup>3</sup> The distance was measured in the space defined by the frequency of accesses to descriptors.

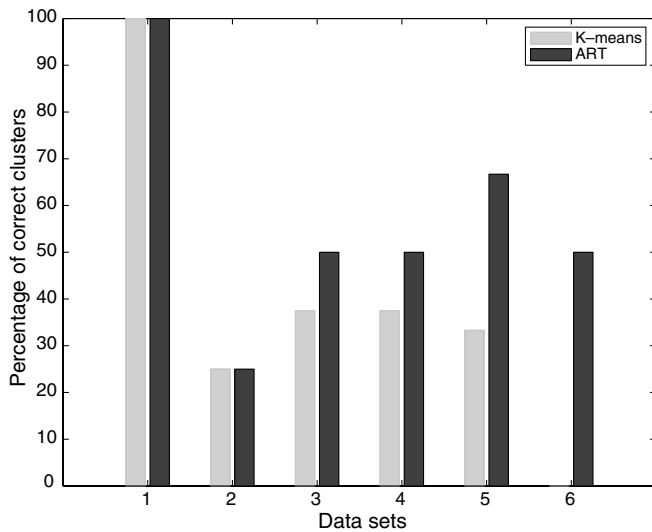


Fig. 5. Percentage of correct clusters found by the ART2 network and by the K-means algorithm. The six artificial data sets are represented in the x-axis.

600 groups, which mix natural clusters or break them up with  
601 unnecessary intermediate clusters.

602 The values of the parameter  $D$  for the six artificial data  
603 sets, using K-means and ART2 networks are shown in  
604 Table 2. This measure is used to determine the goodness  
605 of the clustering together with the percentage of right clusters  
606 found. Apart from the simple data set #1, the empirical  
607 outcomes obtained with ART2 are clearly better, since  $D$   
608 had a smaller value except with data sets #3 and #4; never-  
609 theless, the percentage of correct clusters found by  
610 ART2 with these data sets was considerably higher than  
611 those found by K-means (50% and 66.7% by ART2, and  
612 37.5% in both cases by K-means). Therefore, the conclu-  
613 sion is that K-means works more or less well with a small  
614 number of groups but ART2 best captured the structure of  
615 the known clusters in the tests with artificial data. It is  
616 important to point out that with ART2 the values of  $D$   
617 were very similar for the different data sets (except #1,  
618 which was very simple), indicating the robustness of this  
619 algorithm, since it was able to find correct clusters for dif-  
620 ferent dimensionalities and actual numbers of clusters.  
621 Obviously, an overall assessment of the algorithm must also  
622 take account of the percentage of correct clusters.

Table 2  
Normalized Mahalanobis distance between the actual centres and the correct clusters found by the K-means algorithm and an ART2 network

	K-means	ART2 network
Data set #1	0.0330	0.0330
Data set #2	0.6818	0.2492
Data set #3	0.1498	0.2314
Data set #4	0.2227	0.2747
Data set #5	0.2858	0.2701
Data set #6	–	0.2411

The distances are measured in the descriptors' probability space.

A final test of the algorithms' robustness was carried out  
by analysing the normality of the clusters achieved, given  
that the artificial data sets were generated with multivariate  
Gaussian distributions. For this purpose we can use mea-  
sures of *skewness* and *kurtosis* (Hair, Anderson, Tatham,  
& Black, 1998). Skewness is a measure of symmetry, or  
more precisely, the lack of symmetry. On the other hand,  
kurtosis is a measure of whether the data are peaked or flat  
relative to a normal distribution. A statistical test based on  
skewness and kurtosis values was carried out, testing for a  
normality to a confidence of 95.5%. All clusters found by  
either method for data sets #1 and #2 were consistent with  
normality. However, with the higher dimensional data sets,  
K-means showed a high percentage of non-normal clusters  
of 25% and 50% compared with 8% and 25% for ART2,  
respectively. This indicates that ART2 more closely cap-  
tured normality of the artificial clusters.

#### 4.2. Preliminary clustering of accesses to Infoville XXI

First, a preliminary study was carried out just to know  
the capabilities of the algorithms to find useful and under-  
stable 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–January  
2003) was used. First, the access frequencies of each  
descriptor were analysed to remove those descriptors that  
provided the slightest information. From the remaining  
descriptors, five were selected by Tissat, S.A.<sup>4</sup> as the most  
significant ones: public administration, town councils,  
channels, shopping and entertainment. This resulted in  
1676 users for this study.

The results were analysed in terms of the interpretability  
of the clusters obtained. This was possible because the clus-  
tering was done in a five-dimension space, in which the  
meaning of all the components was known. The clustering  
achieved by K-means was not easy to understand, and the  
clusters did not represent logical behaviours of people,  
indeed. However, the ART2 clustering was quite straight-  
forward, 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 centred between the descriptors "shop-  
ping" and "entertainment". Therefore, it clustered individ-  
uals who mainly accessed the portal for leisure purposes.  
The other cluster was centred between the descriptors  
"public administration" and "town councils", and it also  
presented a small membership to the descriptor "chan-  
nels". Therefore, people clustered in this group clearly  
accessed the portal for administrative purposes. These  
seven clusters demonstrate two important facts: on the  
one hand, ART2 seems to be suitable as a clustering tool

<sup>4</sup> Tissat, S.A. is the company responsible for developing the portal.

675 for this portal; on the other hand, the usefulness of the por-  
 676 tal is clearly demonstrated, since it was basically designed  
 677 to accomplish these two requirements, i.e., to accelerate  
 678 administrative paperwork, and to provide a fast gateway  
 679 for the leisure interests of citizens.

#### 680 4.3. Final clustering and viability of recommendations 681 in *Infoville XXI*

682 Finally, we clustered the data set formed by the users of  
 683 *Infoville XXI* described in Section 2.2. The results of clus-  
 684 tering achieved with artificial data sets, and also the preli-  
 685 minary clustering of accesses to *Infoville XXI* both  
 686 suggest the use of ART2 as clustering tool. However, in  
 687 order to carry out a last comparison we also clustered these  
 688 data set by using K-means. Since the clusters were not  
 689 known in advance, the evaluation described for the artifi-  
 690 cial data sets could not be carried out, nor was it feasible  
 691 to analyse the interpretability of the groups obtained due  
 692 to the high-dimensional space in which the clustering was  
 693 performed. The evaluation of the clustering could be  
 694 assessed by means of analysing the success of the recom-  
 695 mendations based on this clustering. This is an approach  
 696 which can be used not only to evaluate the clustering, but  
 697 also, to study the feasibility of a recommender before its  
 698 actual implementation.

699 Clustering was used to carry out a kind of collaborative  
 700 filtering, i.e., the most likely service of the user group was  
 701 recommended, provided that this service had not yet been  
 702 accessed. Services based on clustering could not be recom-  
 703 mended for the first accesses, since there was not enough  
 704 information to assign users to a certain cluster. Instead,  
 705 the most likely services of the portal were recommended  
 706 for these first accesses, provided that they had not yet been  
 707 accessed.

708 Afterwards, a test was performed to determine whether  
 709 or not users actually click on the recommended object.  
 710 Finally, the success ratio (SR) achieved in the prediction  
 711 of the accessed services by using our methodology (collab-  
 712 orative recommendation based on clustering) was bench-  
 713 marked with that SR obtained by recommending only the  
 714 most likely service of the whole portal that had not yet  
 715 been accessed (Naïve–Bayes recommendation). Therefore,  
 716 the effectiveness of our methodology was measured in  
 717 terms of the improvement in the SR with respect to the  
 718 methodology which did not use clustering. ART2 yielded  
 719 a clustering formed by 12 groups of users, which corre-  
 720 sponded with a vigilance parameter  $\rho = 0.8$ ; slight differ-  
 721 ences in this value led to a considerably different number  
 722 of clusters. Therefore, we considered 12 groups as a natural  
 723 number of clusters for this data set, and hence, we assumed  
 724 12 groups for K-means clustering, as well.

725 The average success ratio (ASR) over the 14,076 acces-  
 726 ses used for the evaluation is benchmarked in Table 3 for  
 727 different values of  $m$  (number of accesses needed to carry  
 728 out a prediction) and  $l$  (depth of the prediction) and for  
 729 ART clustering recommendation and a naïve recommenda-

Table 3

Average success rate, ASR (%) measuring the goodness of service prediction as a preliminary step in the development of a recommender

$m$	$l$	No clustering	ART2 clustering
2	4	6.91	12.84
2	5	10.12	14.57
2	6	13.07	16.48
2	7	16.13	18.74
3	4	3.47	13.73
3	5	7.04	15.11
3	6	10.31	16.81
3	7	13.70	18.97
4	6	7.56	18.06
4	7	11.32	19.94
5	7	8.16	20.94

Prediction with and without clustering is benchmarked for different values of  $m$  and  $l$ .

730 tion. Results with K-means were not included since they  
 731 were similar to naïve recommendations, and much worse  
 732 than those obtained by ART2, as it was expected from  
 733 the results of the previous tests with artificial data sets  
 734 and with a reduced version of the real data set. It can be  
 735 observed that our methodology based on using ART2 clus-  
 736 tering information yields higher ASRs than the methodol-  
 737 ogy that does not take into account clustering information.  
 738 As more accesses are used to cluster, better results are  
 739 obtained; this is expected, since the information gathered  
 740 by the clustering algorithms is more extensive. Besides this,  
 741 the importance of the clustering appears to be more rele-  
 742 vant in the first accesses starting from the  $(m + 1)$ th one;  
 743 as the number of accesses increase, the difference between  
 744 using clustering information or not becomes smaller.  
 745 Therefore, clustering appears to be particularly important  
 746 in the first accesses of the users, when they must be  
 747 attracted in order to establish their loyalty to the portal.

## 748 5. Discussion

749 Recommender systems are one of the most prolific fields  
 750 of research and publication of user modelling. In this work,  
 751 we focus our efforts on recommendation systems for Web  
 752 sites, although their application to other areas is also pos-  
 753 sible with some small changes. A good recommender is  
 754 undoubtedly useful since users can achieve the objects  
 755 searched for in less time, or even better, find something  
 756 interesting that they would not have found by themselves.  
 757 It is also useful for the company which exploits the site,  
 758 since obvious economical profits can be obtained from use-  
 759 ful recommendations. Finally, a good recommender also  
 760 provides an indirect benefit, which is the improvement of  
 761 the Web site.


762 However, the development of such systems is not easy,  
 763 and in addition, it may involve a high economic invest-  
 764 ment. Until now, recommender systems used to be devel-  
 765 oped and then evaluated; in this work, we propose an  
 766 approach which consists of carrying out an evaluation of  
 767 the intended methodology for the design of the recom-  
 768 mender system before being implemented in order to

769 analyse its feasibility and to tune its performance. It opti- 824  
 770 mises predictive accuracy over a range of artificial data 825  
 771 models, before testing the system on retrospective test data. 826  
 772 If this prediction works, then the users have been success- 827  
 773 fully profiled. Besides, it is logical to believe that a recom- 828  
 774 mender system using a similar strategy would work even 829  
 775 better, since attractive recommendations can affect user 830  
 776 behaviour, making the users click on such recommenda- 831  
 777 tions. Therefore, the success obtained with the prediction 832  
 778 could be interpreted as a lower threshold of the success that 833  
 779 can be obtained with a similar recommender system. 834

780 In particular, we have benchmarked a prediction based 840  
 781 on using information about clustering with one that pre- 841  
 782 dicts the most likely service of the portal that has not yet 842  
 783 been accessed by the recommended user. More specifically, 843  
 784 we use clustering of users to classify new users in a certain 844  
 785 group, thus finding out which service will be the most likely 845  
 786 for this user. In order to make the prediction useful from a 846  
 787 recommendation point of view, it is important not to pre- 847  
 788 dict/recommend services already accessed by users, since 848  
 789 these objects are already known by them, and therefore, 849  
 790 they do not provide new information about the portal, 850  
 791 which is one of the most important goals of a recommender 851  
 792 system. 852

793 ART2 and K-means have been benchmarked in some 853  
 794 artificial data sets, which have been obtained by a user 854  
 795 model; the data sets represent different kinds of Web usage 855  
 796 sites. Moreover, these algorithms have also been bench- 856  
 797 marked in a real data set, consisting of accesses to the 857  
 798 Web portal *Infoville XXI*. All these tests show that ART 858  
 799 is a far more suitable technique than the classical K-means. 859

800 The results show that using ART2 clustering informa- 860  
 801 tion provides a much better prediction, showing success 861  
 802 rates which are approximately double the rates obtained 862  
 803 with respect to prediction using K-means clustering or 863  
 804 without any kind of clustering information. Although it 864  
 805 might seem obvious, the authors want to point out that 865  
 806 the users used for clustering are different from the users 866  
 807 used to evaluate the prediction. This demonstrates the 867  
 808 robustness of the clustering achieved, and the relevance 868  
 809 of the information provided by it. Moreover, the percent- 869  
 810 age of success in the recommendation can be considered 870  
 811 as very important and relevant, since typical recommenders 871  
 812 tend to yield percentages of acceptance considerably lower 872  
 813 (Geyer-Schulz & Hashler, 2002), and in addition, these 873  
 814 results should be understood as a lower threshold of the 874  
 815 success that can be obtained with a similar recommender 875  
 816 system, actually. 876

817 Part of the approach proposed in this work is already 877  
 818 implemented in the software iSUM® ( <http://www.isum.com/>), and nowadays, the implementation of all the meth- 878  
 819 odology is being considered. 879  
 820 880

## 821 6. Conclusions

822 A novel approach to evaluate the viability of imple- 881  
 823 menting a recommender in Web portals is proposed. The 882

823 first step of the proposed methodology is to cluster 824  
 824 user data based on simulations in order to ensure that 825  
 825 the collaborative filter is robust across a range of user 826  
 826 models. These results demonstrated the predictive accu- 827  
 827 racy of cluster-based recommender systems using the 828  
 828 ART2 neural network algorithm applied to profiles of 829  
 829 simulated user data. This predictive accuracy then sup- 830  
 830 ports the offer of services that are new to the user. 831  
 831 The results on a retrospective sample of real-world data 832  
 832 show a considerable improvement with respect to either 833  
 833 a prediction based on K-means clustering or a prediction 834  
 834 which does not take into account clustering information, 835  
 835 indicating that the proposed methodology will add value 836  
 836 to the design of cluster-based collaborative recommender 837  
 837 systems for users of the citizen information Web portal 838  
 838 studied. 839

840 The proposed methodology has generic applicability to 840  
 841 other Web portals, which include anticipated growth areas, 841  
 842 for instance, interactive TV, where the user model would 842  
 843 have to be re-estimated. 843

844 Future work will be dedicated to carrying out a follow- 844  
 845 up of real recommendations once these data are available. 845  
 846 This follow-up should be used to improve the recom- 846  
 847 mender system, since feedback of actual recommendations 847  
 848 can be used to adapt the system, potentially with adaptive 848  
 849 on-line profiling. 849

## 850 Acknowledgements

851 This work has been partially supported by the research 851  
 852 project entitled “NDPG (Neuro-Dynamic Programming 852  
 853 Group): Aplicaciones prácticas de programación dinámica 853  
 854 y aprendizaje reforzado en minería web y marketing”, with 854  
 855 reference number GV05/009. 855

## 856 Appendix A. ART2 algorithm

857 Let 857  
 858  $E^k$   $k$ th input pattern 858  
 859  $p$  the dimension of the training examples and proto- 859  
 860 types 860  
 861  $\alpha$  positive number  $\leq 1/\sqrt{p}$  861  
 862  $\beta$  small positive number 862  
 863  $\theta$  normalization parameter, with  $0 \leq \theta \leq 1/\sqrt{p}$  863  
 864  $\rho$  vigilance parameter, with  $0 \leq \rho \leq 1$ . 864

- 865 0. Preprocess all training examples using threshold  $\theta$ . 865
  - 866 0a. Normalize all  $E^k$ . 866
  - 867 0b. Replace every component  $E_j^k$  that is  $\leq \theta$  by 0. 867
  - 868 0c. Renormalize all  $E^k$ . 868
- 869 1. Start with no prototype vectors. 869
- 870 2. Perform iterations until none of the training examples 870  
 871 cause any change in the set of prototype vectors; at this 871  
 872 point quit because stability has been achieved. For each 872  
 873 iteration take the next training example,  $E^k$ , chosen in 873  
 874 cyclic order. 874

- 885 3. Find the prototype  $P_i$  (if any) not yet tried during this  
886 iteration that maximizes  $P_i \cdot E^k$ .  
887 4. Test whether  $P_i$  is sufficiently similar to  $E^k$ :

$$P_i \cdot E^k \geq \alpha \sum_j E_j^k?$$

889

890

891

892

- 4a. If not then:

4aa. Make a new cluster with prototype set to  $E^k$ .

4ab. End this iteration and return to step 2 for the next example.

893

894

895

896

897

898

- 4A. If sufficiently similar, then test for vigilance acceptability:

$$P_i \cdot E^k \geq \rho$$

900

901

902

903

- 4Aa. If acceptable then  $E^k$  belongs in  $P_i$ 's cluster. Modify  $P_i$  to be more like  $E^k$

$$P_i = \frac{(1 - \beta)P_i + \beta E^k}{\|(1 - \beta)P_i + \beta E^k\|}$$

905

906

907

908

909

910

911

and go to step 2 for the next iteration with the next example.

- 4AA. If not acceptable, then make a new cluster with prototype set to  $E^k$  and return to step 2 for the next example.

## 913 References

914 Andersen, J., Larsen, R. S., Giversen, A., Pedersen, T. B., Jensen, A. H., &  
915 Skyt, J. (2000). Analyzing clickstreams using subsessions. Technical  
916 report TR-00-5001, Department of Computer Science, Aalborg  
917 University.

918 Balaguer, E., & Palomares, A. (2003). AI recommendation engine of tissat,  
919 S.A. Internal Technical Report, Tissat, S.A. Valencia, Spain.

920 Banerjee, A., & Ghosh, J. (2002). Characterizing visitors to a Web site  
921 across multiple sessions. In *Proceedings of NGDM'02: National  
922 science foundation workshop on next generation data mining*, Balti-  
923 more, USA.

924 Baudisch, P., & Brueckner, L. (2002). TV scout: lowering the entry barrier  
925 to personalized TV program recommendation. In *Proceedings of the  
926 2nd international conference AH2002* (pp. 58–68), Malaga, Spain.

927 Breese, J. S., Keckerman, D., & Kadie, C. (1998). Empirical analysis of  
928 predictive algorithms for collaborative filtering. In *Proceedings of the  
929 14th annual conference on uncertainty in artificial intelligence* (pp. 43–52).

930 Breslau, L., Cao, P., Fan, L., Phillips, G., & Shenker, S. (1999). Web  
931 caching and zipf-like distributions: evidence and implications. *Pro-  
932 ceedings of INFOCOM 1999*, 1, 126–134.

933 Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2001).  
934 Model-based clustering and visualization and navigation patterns on a  
935 web site technical report MSR-TR-0018, Microsoft Research, Micro-  
936 soft Corporation.

937 Carberry, S. (2001). Techniques for plan recognition. *User Modeling and  
938 User Adapted Interaction*, 11, 31–48.

939 Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architec-  
940 ture for a self-organizing neural pattern recognition machine. *Com-  
941 puter Vision, Graphics and Image Processing*, 1137, 54–115.

942 Carpenter, G. A., & Grossberg, S. (1991). *ART2: Self-organization of  
943 stable category recognition codes for analog input patterns. Pattern  
944 recognition by self-organizing neural networks*. MIT Press.

Cosley, D., Shyong, K. L., Albert, I., Konstan, J., & Riedl, J. (2003). Is  
945 seeing believing? How recommender system interfaces affect users'  
946 opinions. In *Proceedings of the ACM SIGCHI conference on human  
947 factors in computing systems, 2003*. Minneapolis, USA.  
948

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*.  
949 John Wiley and Sons.  
950

Fu, Y., Shandu, K., & Shih, M. (1999). Fast clustering of web users based  
951 on navigation pattern. In *Proceedings of SCI'99/ISAS'9*, Orlando,  
952 USA.  
953

Geyer-Schulz, A., & Hashler, M. (2002). Evaluation of recommender  
954 algorithms for an Internet information based on simple association. In  
955 *Proceedings of WEBKDD'02* (pp. 110–114), Edmonton, Canada.  
956

Ghosh, J., Strehl, A., & Meregu, S. (2002). A consensus framework  
957 for integrating distributed clusterings under limited knowledge  
958 sharing. In *Proceedings of NGDM'02: National science foundation  
959 workshop on next generation data mining* (pp. 99–108), Baltimore,  
960 USA.  
961

Hair, J. F., Anderson, R. E., Tatham, R. L., & Black, W. C. (1998).  
962 *Multivariate data analysis* (5th ed.). Upper Saddle River: Prentice Hall.  
963

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). Recommending  
964 and evaluating choices in a virtual community of use. In *CHI'95:  
965 Conference proceedings on human factors in computing systems* (pp.  
966 194–201), Denver, USA.  
967

Kim, Y., Ok, S., & Woo, Y. (2002). A case-based recommender system  
968 using implicit rating techniques. In *Proceedings of the 2nd international  
969 conference AH2002* (pp. 522–526), Malaga, Spain.  
970

Kohonen, T. (1997). *Self-organizing maps* (2nd ed.). Berlin: Springer-  
971 Verlag.  
972

Lang, K. (1995). Newsweeder: learning to filter news. In *Proceedings of the  
973 12th international conference on machine learning* (pp. 331–339), Lake  
974 Tahoe, USA.  
975

Lee, M., Choi, P., & Woo, Y. (2002). A hybrid recommender system  
976 combining collaborative filtering with neural networks. In *Proceedings  
977 of the 2nd international conference AH2002* (pp. 531–534), Malaga,  
978 Spain.  
979

~~Martin, J. D. (2003). A pseudo-supervised approach to improve a  
980 recommender based on collaborative filtering. In *Proceedings of  
981 the 9th international conference UM2003* (pp. 429–431), Johnstown,  
982 USA.  
983~~

Martin, J. D., Palomares, A., Balaguer, E., Soria, E., Gomez, J., &  
984 Soriano, A. (2006). Studying the feasibility of a recommender in a  
985 citizen web portal based on user modeling and clustering algorithms.  
986 *Expert Systems with Applications*, 30, 299–312.  
987

McNee, S. M., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). Interfaces  
988 for eliciting new user preferences in recommender systems. In  
989 *Proceedings of the 9th international conference UM2003* (pp. 178–  
990 187), Johnstown, USA.  
991

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994).  
992 An open architecture for collaborative filtering of netnews. In  
993 *Proceedings of the conference on computer supported cooperative work*  
994 (pp. 175–186), Chapel Hill, USA.  
995

Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender systems in E-  
996 commerce. In *Proceedings of the first ACM conference on electronic  
997 commerce EC'99* (pp. 158–166), Denver, USA.  
998

Shardanand, U., & Maes, P. (1995). Social information filtering:  
999 algorithms for automating “word of mouth”. In *CHI'95: Conference  
1000 proceedings on human factors in computing systems* (pp. 210–217),  
1001 Denver, USA.  
1002

Su, Z., Ye-Lu, Q. Y., & Zhang, H. J. (2000). WhatNext: a prediction  
1003 system for Web requests using N-gram sequence models. In *WISE  
1004 2000 proceedings: 1st international conference on web information  
1005 systems engineering* (pp. 214–221), Hong Kong.  
1006

Zhou, D., Weston, J., Gretton, A., & Schölkopf, B. (2003). Ranking on  
1007 data manifolds. Technical report num. TR-113. Tuebingen, Germany:  
1008 Max Planck Institute for Biological Cybernetics.  
1009

Zukerman, I., & Albrecht, D. W. (2001). Predictive statistical models for  
1010 user modeling. *User Modeling and User Adapted Interaction*, 11,  
1011 5–18.  
1012  
1013