Applied Soft Computing 11 (2011) 793–806

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

NEWER: A system for NEuro-fuzzy WEb Recommendation

G. Castellano, A.M. Fanelli, M.A. Torsello*

University of Bari, Department of Informatics, Via Orabona, 4, 70126 Bari, Italy

ARTICLE INFO

Article history:
Received 24 October 2008
Accepted 31 December 2009
Available online 14 January 2010

Keywords:
Neuro-fuzzy systems
User profiling
Web recommendation
Web personalization
Web usage mining

ABSTRACT

In the era of the Web, there is urgent need for developing systems able to personalize the online experience of Web users on the basis of their needs. Web recommendation is a promising technology that attempts to predict the interests of Web users, by providing them with information and/or services that they need without explicitly asking for them. In this paper we propose NEWER, a usage-based Web recommendation system that exploits the potential of Computational Intelligence techniques to dynamically suggest interesting pages to users according to their preferences. NEWER employs a neuro-fuzzy approach in order to determine categories of users sharing similar interests and to discover a recommendation model as a set of fuzzy rules expressing the associations between user categories and relevances of pages. The discovered model is used by a online recommendation module to determine the list of links judged relevant for users. The results obtained on both synthetic and real-world data show that NEWER is effective for recommendation, leading to a quality of the generated recommendations comparable and often significantly better than those of other approaches employed for the comparison.

1. Introduction

During the past few years, the World Wide Web has become the biggest and the most popular way of communication and information dissemination. Everyday, the Web grows by roughly millions of electronic pages, adding to the hundreds of millions pages already on-line. Because of its rapid and chaotic growth, the resulting network of information lacks of organization and the structure of Web sites becomes more and more complex. When searching and browsing the Web, users are very often overwhelmed by huge amount of information and are faced with a big challenge to find the most relevant information in right time. As a consequence, users often feel disoriented and get lost in that information overload that continues to expand.

Web personalization represents one of the most promising and potent remedies against the problem of information overload. Today, the need of predicting the user preferences and customizing the interactions on a Web site according to the implicit/explicit interests and desires of users is more than ever evident. Besides, the ability of a site to engage visitors at a deeper level, and to successfully guide them to useful and pertinent information, is now viewed as one of the key factors for the site's ultimate success.

Personalization plays a fundamental role in many contexts and, dependently on the context, it may be used to achieve several goals, ranging from increasing customer loyalty in e-commerce sites to enable better search providing results satisfying user needs. Recommendation systems are one of the major examples of personalization systems. Such systems have shown to greatly help Web users in navigating the Web, locating relevant and useful information, and receiving dynamic recommendations from Web sites on possible products or services that match their interests. To build Web recommendation systems, the Web usage mining (WUM) methodology is one of the main approaches used in the literature. WUM involves the application of data mining and machine learning techniques to discover usage patterns (or build user models) through the analysis of Web users' historical navigational activities. These models may be properly exploited to realize the different personalization functions.

In this way, in the design of a Web recommendation system based on the WUM methodology, three main phases may be distinguished [10]:

- *Web data preprocessing*: usage data are collected and preprocessed to identify user sessions.
- *Knowledge discovery*: useful usage patterns and recommendation models are discovered from preprocessed data.
- *Recommendation*: the discovered models are exploited to deliver intelligent recommendations.

In the process of knowledge discovery, different facets have to be addressed in order to mine models useful for the determination of interesting and accurate recommendations. In fact, the Web is a complex and heterogeneous network of interconnected components. Moreover, Web data are highly characterized by vagueness and imprecision. Their fuzzy and uncertain nature gives rise to the necessity of relevant intelligent techniques able to process the dif-

* Corresponding author. Tel.: +39 0805442456; fax: +39 0805443196.
E-mail address: torsello@di.uniba.it (M.A. Torsello).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.12.040

ferent kinds of Web uncertainty that cannot be heavily processed through the traditional and precise techniques. Due to their characteristics, Computational Intelligence (CI) [18] techniques reveal to be appropriate to deal with this uncertainty and to develop Web-based applications tailored to user preferences. The main reason behind this success seems to be the synergy resulting from CI paradigms, such as fuzzy logic, neural networks and genetic algorithms. Rather than being competitive, each of these computing paradigms provides complementary reasoning and searching methods that allow the use of domain knowledge and empirical data to solve complex problems. On the basis of these considerations, CI techniques have been combined together in different ways leading to several hybrid schemes which, exploiting the strengths of each involved computing paradigm, can achieve high intelligence degrees.

This paper[1] is intended to propose NEWER (NEuro-fuzzy WEb Recommendation), a Web recommendation system that exploits CI techniques to dynamically suggest users interesting links according to their interests. In NEWER, a neuro-fuzzy methodology is implemented in order to discover interesting user navigational patterns and to derive a recommendation model useful for the suggestion of links considered relevant for the users. According to the WUM methodology, in the proposed system, the interests of users are implicitly derived in form of user categories by analyzing the usage data stored by the Web server in log files. Then, on the basis of the derived knowledge about user interests, a recommendation model is discovered via CI techniques to generate intelligent suggestions.

Specifically, as concerns the identification of user categories, in NEWER, a fuzzy clustering technique is employed. The use of this technique enables the generation of overlapping clusters, so that a user can belong to more than one category capturing in this way the overlapping interests of users.

To create the recommendation model which is exploited to provide intelligent predictions about Web pages to be suggested, a neuro-fuzzy approach is employed. This hybrid approach permits to exploit the learning capabilities of neural networks to derive a recommendation model expressed in a comprehensible form, as a set of fuzzy rules which can be easily understood by humans.

The rest of the paper is organized as follows. Section 2 provides a review of works that exploit the WUM methodology in the process of knowledge discovery from Web data. In Section 3, the architecture of the NEWER system is presented. Section 4 describes the step of log file preprocessing. In Section 5, the knowledge discovery step is detailed and in section 6, the ultimate recommendation step is described. Section 7 provides results obtained by testing the NEWER system both on synthetic data and real world data. A comparison between NEWER and other recommendation approaches is presented in Section 8. Finally, in Section 9 some conclusions and future research directions are drawn.

2. Review of related usage mining approaches

A WUM methodology provides a complete process for the extraction of models from usage data encoding the behavior and the interests of users. These models may be automatically exploited by a personalization system to personalize its services.

In this section, we give some examples of personalization systems that exploit techniques underlying the WUM methodology for mining knowledge from Web usage data.

Analog [20] was one of the pioneer personalization systems based on the WUM methodology. In such a system, the mining process is organized into two main components performed offline and online with respect to the server activity. In the offline component, past user activity stored in log files is processed by a geometrical clustering approach to create clusters of user sessions. Then, the online component creates active user sessions which are classified into one of the clusters previously identified. This permits to identify pages related to those in the active session and to return the requested page with the list of related documents.

Mobasher et al. [9] propose a usage-based Web personalization system taking into account both the offline tasks related to the mining of usage data, and the online process of automatic Web page customization based on the mined knowledge. In particular, usage mining tasks involved the discovery of association rules and the derivation of URL clusters. Once the mining tasks have been accomplished, the frequent itemsets and the URL clusters are used to provide dynamic recommendations to users based on their current navigational activity.

A WUM approach has been also exploited in SiteHelper [14], a system that has been designed to adapt Web pages to the user needs. In this system, usage mining techniques are employed to build a set of rules that represent the user interests. The discovered rules are used by the system to recommend new or update Web pages to the users according to their interests.

WUM techniques have been employed in KOINOTITES [17] in order to customize information to the needs of individual users. More specifically, such system identifies user communities which model groups of visitors in a Web site having similar interests and navigational behavior. These communities are exploited by the administrator of the site to improve the organization of the site or as input to a personalization system to dynamically make recommendations to Web users. The mining component of KOINOTITES includes four steps that perform the main functions: data preprocessing, session identification, pattern recognition and knowledge presentation. In pattern discovery, usage models of users are extracted by a variation of *Cluster Mining*, a simple graph-based clustering algorithm.

SUGGEST [2] is another example of WUM system designed to provide useful information to make easier the Web user navigation. SUGGEST adopts a two-level architecture composed by a offline creation of historical knowledge and a online engine that understands user behavior. It creates clusters of related pages based on user past activity and then classifies new users by comparing pages in their active sessions with pages inside the clusters created. A set of suggestions is then obtained for each request.

In [12], Nasraoui and Petenes present an intelligent Web recommendation system based on WUM to discover useful knowledge about user access patterns. In the mining process, log files are processed to identify user sessions. Then, user profiles are extracted by categorizing the identified sessions through the Hierarchical Unsupervised Niche Clustering algorithm. Finally, in the recommendation engine based on a fuzzy approximate reasoning, the mined knowledge is exploited to determine a set of links to be recommended.

Albanese et al. [1] proposed a Web personalization system based on a usage mining strategy consisting in two phases. In the first phase, a fuzzy unsupervised clustering algorithm is used to classify users by deriving groups of users appearing to be similar. The second phase performs the reclassification of users by taking advantage from the interactions of each user with the Web site.

In [8], the authors proposed a Web recommendation system based on a maximum entropy model. In such a system, different levels of knowledge about the user navigational behavior are combined to generate recommendations for new users. This knowledge includes page-level clickstream statistics about the past navigations of users with the aggregate usage patterns discovered through the WUM methodology.

[1] A preliminary version of this paper has appeared in the Proceeding of the 1st International Conference on the Application of Digital Information and Web Technologies (ICADIWT 2008).

All these systems are examples of personalization systems that employ the WUM methodology in the process of knowledge extraction from large amount of Web data. In such systems, different data mining techniques are used to discover models useful for the generation of intelligent recommendations. In the following sections, we describe the NEWER system, a usage-based system that adopts a new approach for Web recommendation based on a profitable combination of different CI techniques for the discovery of a recommendation model. In particular, NEWER employs a neuro-fuzzy approach to derive a set of recommendation rules in the form of fuzzy rules useful for the online suggestion of interesting links to the current users.

3. The NEWER system

According to the general scheme of a usage-based Web personalization system, three different modules can be distinguished in the proposed system:

- *Log file preprocessing*: usage data stored in access log files are analyzed, cleaned and filtered in order to extract user sessions and models of user behaviors representing the basic structures which encode the access patterns exhibited by the users during navigation.
- *Knowledge discovery*: a number of user categories characterizing the common interests of groups of users are derived by a fuzzy clustering process. Then, a recommendation model is generated via a neuro-fuzzy learning strategy for establishing the associations between user categories and pages to be recommended.
- *Recommendation*: interesting pages are dynamically suggested to the current user by exploiting the previously discovered recommendation model. Specifically, when a user requests a new page, his current partial session is matched with the session categories previously identified and derives the degrees of relevance for each page by means of a fuzzy inference process.

As illustrated in Fig. 1, these modules are organized into two main macro-modules:

- *a offline module* that includes the first two modules to extract a recommendation model from Web usage data;
- *a online module* that performs the effective recommendation task.

In the following, all the tasks involved in each module of the NEWER system are described in details.

4. Log file preprocessing

The first task performed by NEWER consists in the collection of usage data included into access log files stored by the Web server during the interactions of users with the site. Such files record in chronological order all the information about the accesses of users to the pages of the site, including the user IP address, the date and time of the request, the method of the request (GET, POST, HEAD, etc.), the name of the requested resource (URL), etc. Starting from these information, the characteristics of the browsing behavior of users and, hence, their interests can be captured. For these reasons, server log files represent an invaluable source of usage data which can be conveniently exploited to discover interesting navigational patterns encoding the interests of users.

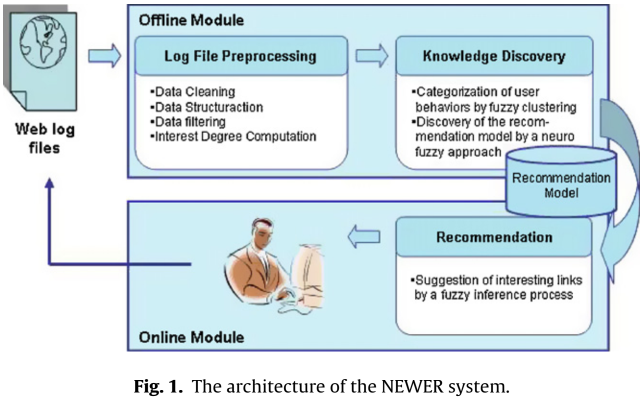

Offline Module
Web log files
Log File Preprocessing
•Data Cleaning
•Data Structuraction
•Data filtering
•Interest Degree Computation
Knowledge Discovery
•Categorization of user behaviors by fuzzy clustering
•Discovery of the recommendation model by a neuro fuzzy approach
Recommendation Model
Recommendation
•Suggestion of interesting links by a fuzzy inference process
Online Module

Fig. 1. The architecture of the NEWER system.

Once access log files have been retrieved, data contained in such files are preprocessed in order to identify user sessions and create models of user navigational behaviors.

This process, known as sessionization [6], is done by identifying the sequence of pages accessed by each user during a certain period of time. The task of log data preprocessing is executed through the following four steps:

- *Data Cleaning* that removes redundant and useless records contained in the log file so as to retain only information concerning accesses to resources of the Web site (typically Web pages).
- *Data Structuration* that groups the significant requests into user sessions. Each user session contains the sequence of pages accessed by the same user during a limited time period.
- *Data Filtering* that selects only significant resources accessed in the Web site. To this aim, the least visited resources as well as the most visited ones, are removed.
- *Interest Degree Computation* that uses information about accessed pages to create a model of the visitor behavior by evaluating a degree of interest of each user for each accessed page.

In the following, each step involved in the preprocessing of log files is detailed.

4.1. Data cleaning

Data cleaning is intended to clean Web log files by deleting irrelevant and useless records in order to retain only usage data that can be effectively exploited to model user navigational behavior. Since Web log files record all user interactions, they represent a huge and noisy source of data, often comprising an high number of unnecessary records. By removing useless data, the size of these files can be reduced in order to use less storage space and to facilitate the upcoming steps. Of course, the choice of which log data to be removed depends on the ultimate goal of the Web personalization system. In this case, since the goal of the NEWER system is to offer personalized dynamic links to the visitors, only explicit requests that actually represent user actions have to be kept. Starting from these considerations, data cleaning removes from log data the following requests:

- *Requests with access method different from "GET"*. Generally, such kind of requests do not refer to explicit requests of users but they often concern with CGI accesses, properties of the Server, visits of robots, etc. Hence, requests containing a value different from "GET" in the field of the access method are considered non-significant and, consequently, they are removed from the log file.
- *Failed and corrupted requests*. These requests are represented by records containing a HTTP error code. A status with value of 200 indicates a succeeded request. All requests having a status code with value different from 200 correspond to failed requests (e.g. a status of 404 indicates that the requested file was not found at the expected location), and hence, they have to be eliminated. In addition, corrupted lines with missing values in some fields are

deleted in order to clean log files from incomplete information.

- *Requests for multimedia objects*. Due to the model underlying the HTTP protocol, a separate access request is executed for every file, image, multimedia object embedded in the requested Web page. As a consequence, when a user requests a Web page, several log entries may often be produced: one for each file automatically downloaded without an explicit request of the same user. These implicit requests can be easily identified since they are characterized by a particular URL name suffix, such as gif, jpeg and jpg. Keeping or removing requests for multimedia objects depends on the kind of Web site to be personalized and on the purpose which the system must achieve. In general, these requests do not represent the effective browser activity of the user visiting the site, hence they are deemed redundant and are removed. In other cases, eliminating requests for multimedia objects may cause a loss of useful information. The decision upon retaining or removing these entries is left to the analyst, who can select the suffixes to be removed in an apposite panel of the data cleaning module.
- *Requests originated by Web robots*. Log files may contain a number of records corresponding to requests originated by Web robots. Web robots (also known as Web crawlers or Web spiders) are programs that automatically download complete Web sites by following every hyperlink on every page within the site in order to update the index of search engine. Requests created by Web robots are not considered usage data and, consequently, have to be removed. Two different heuristics are implemented to identify requests deriving from Web robots. Firstly, all records containing the name “robots.txt” in the requested resource name (URL) are identified and straightly removed. The second heuristic is based on the fact that the crawlers retrieve pages in an automatic and exhaustive manner, so they are characterized by a very high browsing speed (intended as total number of pages visited/total time spent to visit those pages). Hence, for each different IP address the browsing speed is calculated and all requests with this value exceeding a threshold (pages/second) are regarded as made by robots and are consequently removed. The value of the threshold is established by analyzing the browser behavior arising from the considered log files.

Once data cleaning has been performed, only requests for relevant pages are kept. Formally, the set of all distinct pages requested from the Web site under analysis is defined as $P = \{p_1, p_2, \ldots, p_{N_P}\}$.

4.2. Data structuration

Data structuration step is aimed to group the unstructured requests remaining in log data into user sessions. A user session is defined as a limited set of pages accessed by the same user within a particular visit. Identifying user sessions from the log data is a difficult task because many users may use the same computer and the same user may use different computers. Hence, one main problem is how to identify the user. For Web sites requiring user registration, the log file contains the user login that can be used for user identification. In our case, the user login is not available. Thus, a user is simply identified from the IP address, i.e. each IP address is considered as a different user (being aware that an IP address might be used by several users). The set of all the users (i.e. IP addresses) that have accessed that Web site is indicated with $\mathbf{U} = \{u_1, u_2, \ldots, u_{N_U}\}$. According to other works proposed in the literature [15,16], we exploit a time-based method to identify sessions. Precisely, we consider a user session as the set of accesses originating from the same user within a particular time period. Such time period is defined by considering a maximum elapsed time $\Delta t_{\max}$ between two consecutive accesses. Moreover, to better handle particular situations which might occur (such as users accessing several times to the same page due to slow connections or intense network traffic), a minimum elapsed time $\Delta t_{\min}$ between two consecutive accesses is also fixed. Formally, a user session is defined as a triple:

$$\mathbf{s}_i = \left\langle u_i, t_i, \mathbf{p}_i \right\rangle \tag{1}$$

where $u_i \in U$ represents the user identifier, t_i is the access time of the whole session, $\mathbf{p}_i$ is the set of all pages (with corresponding access information) requested during the ith session. Namely:

$$\mathbf{p}_i = \left\langle (p_{i1}, t_{i1}, N_{i1}), (p_{i2}, t_{i2}, N_{i2}), \ldots, \left(p_{i,n_i}, t_{i,n_i}, N_{i,n_i}\right) \right\rangle \tag{2}$$

with $p_{ij} \in P$, where N_{ik} is the number of accesses to page p_k during the ith session and t_{ik} is the access time to that page satisfying the following:

$$t_{i,k+1} \geq t_{i,k} \quad \text{and} \quad \Delta t_{\min} < t_{i,k+1} - t_{i,k} < \Delta t_{\max}$$

Summarizing, after data structuration, a collection of N_S sessions $\mathbf{s}_i$ is identified from the log data. The set of all identified sessions is denoted by $\mathbf{S} = \left\langle \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N_S} \right\rangle$.

4.3. Data filtering

Once user sessions have been identified, a data filtering step is performed. In this step, we execute a page filtering and a session filtering in order to retain only the most visited pages and the most significant user sessions. More precisely, in the page filtering process the following requests are removed:

- Requests for very low support pages, i.e. requests to pages which do not appear in a sufficient number of sessions.
- Requests for very high support pages, i.e. requests to pages which appear in nearly all sessions.

Formally, we consider, for each page p_j, the number NS_j of different sessions that required it and compute the quantity $NS = \max_{j=1,\ldots,N_P} NS_j$. Then, we define a threshold ϵ whose value is a low percentage of NS. Very low support filtering consists in removing all pages that satisfy $NS_j < \epsilon$. Very high support filtering removes all resources such that $NS_j > NS - \epsilon$. This type of support-based filtering is useful to eliminate pages having minimal knowledge value for the purpose of modeling the visitor behavior. In addition, the data filtering module eliminates all the user sessions that comprise only very-low-support pages. In this way, the size of data is even more reduced.

Finally, session filtering process removes all user sessions that include a very low number of visited pages. The threshold used for session filtering is fixed as a very small percentage of the total number of visited pages N_P.

4.4. Interest degree computation

Both the statistics and the user sessions are used in this step to create a model of the user interest. The most commonly used methods to evaluate user interest about pages is by counting page accesses or “hits”. However, this is not sufficient. Access counts, when considered alone, can be misleading metrics. The time collected for each successive request can give interesting clues regarding the user interest by evaluating the amount of time spent by users on each page. This approach seems reasonable, since it tends to weight content pages higher. However, a long access can completely obscure the importance of other relevant pages. Another possibility is to define interest degrees by the number of times a page was visited during the navigation. The metric adopted in NEWER expresses the interests degree for each page the user accessed during her/his navigation as a function of two variables: the overall time the user spends on the page during its visit and the frequency of accesses to the page within the session. Formally, given a page $p_{ij} \in P$ accessed in the ith user session, with access time t_{ij}, the following measure is used to estimate the interest degree:

$$IG_{ij} = f_{ij} \cdot \frac{t_{ij}}{t_i} \tag{3}$$

where $f_{ij} = N_{ij} / \sum_{k=1}^{n_i} N_{ik}$ is the frequency of accesses to the jth page within session $\mathbf{s}_i$.

Since the number of Web pages accessed by different users may vary, any two different session vectors $\mathbf{s}_i \neq \mathbf{s}_l$ may have different dimension, i.e. $n_i \neq n_l$. In order to create a homogeneous model for all visitors, we need to create vectors with the same number of components. Let n be the total number of pages required in all the identified user sessions, i.e. $n = \max_{i=1,\ldots,n_s} n_i$. Then, we model the browsing behavior of a user $u_i \in U$ through a vector $\mathbf{b}_i = (b_{i1}, b_{i1}, \ldots, b_{in})$ where

$$b_{ij} = \begin{cases} IG_{ij} & \text{if page } p_j \text{ is accessed in session } s_i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The outcome of log file preprocessing is represented by a $n \times m$ matrix $\mathbf{B} = [b_{ij}]$ where n and m are respectively the final number of retained sessions (users) and the final number of considered pages. In this matrix, named behavior matrix, each entry b_{ij} represents the interest degree of the ith user for the jth page.

Based on the behavior matrix, users with similar preferences can be successively clustered together into user categories, as described in the following sections.

5. Knowledge discovery

Once the log data preprocessing has been completed, the successive step of NEWER is knowledge discovery which includes the following tasks:

- the categorization of user behaviors to identify patterns describing the common interests and the navigational behavior of similar users (i.e. user categories);
- the discovery of a knowledge base (recommendation model) which contains a set of recommendation rules.

More precisely, based on the behavior matrix $\mathbf{B}$, a fuzzy clustering process is applied in order to identify user categories. Each user category will include users exhibiting a common browsing behavior and hence similar interests. Next, the user sessions and the derived categories are exploited in order to extract a knowledge base which is useful for the effective online personalization step. Knowledge extraction is performed through the application of a neuro-fuzzy model which provides a fuzzy rule base, where each rule represents the associations between user behaviors and pages to be recommended.

In the next sections, the activities of categorization of user behaviors and discovery of recommendation model are described in more detail.

5.1. Categorization of user behaviors

Categorization of user behavior vectors is devoted to identify a set of aggregate user categories characterizing the similar interests of users exhibiting common browsing behavior.

To identify user categories, NEWER applies CARD+, a fuzzy clustering algorithm that we proposed in [5] as an improved version of the Competitive Agglomeration Relational Data (CARD) algorithm [11]. CARD+ overcomes CARD in the ability to automatically categorize the available data into an optimal number of clusters starting from an initial random number. Since the actual number of user categories visiting a Web site is not known in advance, clustering algorithms capable of automatically determining the number of clusters are especially required to perform this task. Actually, we experimented that CARD is sensitive to the initial number of clusters $C_{\max}$ since it often provides different final partitions for different values of $C_{\max}$ and thus failing in finding the actual number of clusters buried in data. Indeed, we observed that CARD produces redundant partitions, with clusters having a high overlapping degree (very low inter-cluster distance). To overcome this limitation, we developed CARD+ by adding a post-clustering process to the CARD algorithm in order to remove redundant clusters.

As common relational clustering algorithms, CARD+ partitions the available data by working on relational data, i.e. data that quantify the relation between each pair of objects. Typically, the relation expresses the similarity degree existing between two objects. In NEWER, to capture the similarity between two generic behavior vectors, CARD+ has been equipped with a new fuzzy similarity measure that we proposed in [4]. Specifically, each behavior vector is modeled as a fuzzy set and the similarity between two behavior vectors is expressed as the similarity between the corresponding fuzzy sets.

To do so, the user behavior matrix $\mathbf{B}$ is converted into a matrix $\mathbf{F} = [\mu_{ij}]$ which expresses the interest degree of each user for each page in a fuzzy way. A very simple characterization of the matrix $\mathbf{F}$ is provided as follows:

$$\mu_{ij} = \begin{cases} 0 & \text{if} \quad b_{ij} < ID_{\min} \\ \dfrac{b_{ij} - ID_{\min}}{ID_{\max} - ID_{\min}} & \text{if} \quad b_{ij} \in [ID_{\min}, ID_{\max}] \\ 1 & \text{if} \quad b_{ij} > ID_{\max} \end{cases} \tag{5}$$

where $ID_{\min}$ is a minimum threshold for the interest degree under which the interest for a page is considered null, and $ID_{\max}$ is a maximum threshold of the interest degree, after which the page is considered surely preferred by the user.

Starting from this fuzzy characterization, the rows of the new matrix $\mathbf{M}$ are interpreted as fuzzy sets defined on the set of Web pages. Each fuzzy set μ_i is related to a user behavior vector $\mathbf{b}_i$ and it is simply characterized by the following membership function:

$$\mu_i(j) = \mu_{i,j} \quad \forall j = 1, 2, \ldots, m \tag{6}$$

In this way, the similarity of two generic users is intuitively defined as the similarity between the corresponding fuzzy sets (rows of $\mathbf{F}$).

Similarity of fuzzy sets can be evaluated in different ways [19,21]. One of the most common measures to evaluate similarity between two fuzzy sets is the following:

$$\sigma(\mu_1, \mu_2) = \frac{|\mu_1 \cap \mu_2|}{|\mu_1 \cup \mu_2|} \tag{7}$$

According to this measure, the similarity between two fuzzy sets is given by the ratio of two quantities: the cardinality of the intersection of the fuzzy sets and the cardinality of the union of the fuzzy sets. The intersection of two fuzzy sets is defined by the minimum operator:

$$(\mu_1 \cap \mu_2)(j) = \min\left\{\mu_{\mathbf{b}_1}(j)\mu_{\mathbf{b}_2}(j)\right\} \tag{8}$$

The union of two fuzzy sets is defined by the maximum operator:

$$(\mu_1 \cup \mu_2)(j) = \max\left\{\mu_{\mathbf{b}_1}(j)\mu_{\mathbf{b}_2}(j)\right\} \tag{9}$$

The cardinality of a fuzzy set (also called "σ-count") is computed by summing up all its membership values:

$$|\mu| = \sum_{j=1}^{m} \mu(j) \tag{10}$$

Summarizing, the similarity between two user behavior vectors is defined as follows:

$$Sim(\mathbf{b}_1, \mathbf{b}_2) = \frac{\sum_{j=1}^{m} \min\left\{\mu_{\mathbf{b}_1,j}, \mu_{\mathbf{b}_2,j}\right\}}{\sum_{j=1}^{m} \max\left\{\mu_{\mathbf{b}_1,j}, \mu_{\mathbf{b}_2,j}\right\}}. \tag{11}$$

Similarity values are mapped into the similarity matrix $\mathbf{Sim} = [Sim_{ij}]_{i,j=1,\ldots,n}$ where each component Sim_{ij} expresses the similarity value between the user behavior vectors $\mathbf{b}_i$ and $\mathbf{b}_j$ calculated by using the fuzzy similarity measure. Starting from the similarity matrix, the dissimilarity values are simply computed as $Diss_{ij} = 1 - Sim_{ij}$, for $i, j = 1, \ldots, n$. These are mapped in a $n \times n$ matrix $\mathbf{R} = [Diss_{ij}]_{i,j=1,\ldots,n}$ representing the relation matrix.

Algorithm 1. The CARD+ Algorithm

```
1: Fix the maximum number of clusters $C = C_{\max}$ $(2 \le C \le n)$; Set the threshold $\epsilon_1$; Initialize $k = 0$; $\beta = 0$; the initial partition matrix $\mathbf{U}^{(0)}$; $N_i = \sum_{j=1}^{n} u_{ij}$, $1 \le i \le C$; the initial $C$ prototype vectors $\mathbf{PV} = \left\{\mathbf{pv}_1, \mathbf{pv}_2, \ldots, \mathbf{pv}_C\right\}$
2: Repeat
    (2.1) Compute membership vectors
          $\mathbf{z}_i = \frac{(\mathbf{u}_{i1}, \ldots, \mathbf{u}_{in})^t}{\sum_{j=1}^{n} u_{ij}}, \quad 1 \le i \le C;$
    (2.2) Compute $d_{ik} = (\mathbf{Rz}_i)_k - \mathbf{z}_i\mathbf{Rz}_i/2$;
    (2.3) If ($d_{ik} < 0$ for any $i$ and $k$) Then
            (i) Compute $\Delta\beta = \max_{ik}\left\{-2d_{ik}/||\mathbf{z}_i - \mathbf{e}_k||^2\right\}$;
            (ii) Update $d_{ik} = d_{ik} + \left(\Delta\beta/2\right) * ||\mathbf{z}_i - \mathbf{e}_k||^2$ for $1 \le i \le C$ and $1 \le k \le n$;
            (iii) Update $\beta = \beta + \Delta\beta$;
    (2.4) Update $\alpha(k) = \eta_0 e^{-k/\tau} \frac{\sum_{i=1}^{C}\sum_{j=1}^{n}(u_{ij})^2 d(\mathbf{b}_j, \beta_i)}{\sum_{i=1}^{C}\left[\sum_{j=1}^{n} u_{ij}\right]}$ where $\eta_0$ is an exponential decay and $\tau$ is the time constant;
    (2.5) Update $\mathbf{U}^{(k)} = \mathbf{U}^{FCM} + \mathbf{U}^{Bias}$ where
          $\mathbf{U}^{FCM} = \frac{1/d(x, \mathbf{PV})}{\sum_{k=1}^{C} \frac{1}{d(x, \mathbf{PV}_k)}}$
          and
          $\mathbf{U}^{Bias} = \frac{\alpha}{d\left(x, \beta\right)(n - \bar{n})}$
            where
          $\bar{n} = \frac{\sum_{k=1}^{C} 1/d(x, \mathbf{pv}_k) N_k}{\sum_{k=1}^{C} 1/d(x, \mathbf{pv}_k)}$
    (2.6) Compute $N_i = \sum_{j=1}^{n} u_{ij}$;
    (2.7) If ($N_i < \epsilon_1$) Then Discard $i$th cluster and Update $C$ and $\mathbf{PV}$;
    (2.8) k=k+1;
          Until (Membership stabilize)
3: Create clusters
    $\chi_c = \left\{\mathbf{b}_i \in \mathbf{B} | d_{ci} < d_{ki} \forall c \neq k\right\}, \quad 1 \le c \le C;$
4: Compute prototype vectors
    $v_{cj} = \frac{\sum_{\mathbf{b}_i \in \chi_c} b_{ij}}{|\chi_c|}, \quad j = 1, \ldots, m$
5: Compute the inter-cluster distance values $\mathbf{D} = \left[D_{ij}\right]_{i,j=1,\ldots,C}$;
6: Compute the average value $\epsilon$ of $D_{ij}$, $i, j = 1, \ldots, C$, $i \neq j$;
7: If ($D_{ij} < \epsilon$ for any $i$ and $j$) Then
    (7.1) Join clusters $\mathbf{v}_i$ and $\mathbf{v}_j$;
    (7.2) Update $C$;
    (7.3) Update cluster prototypes $\mathbf{v}_i$, $i = 1, \ldots, C$;
    (7.4) Update partition matrix $U$;
    (7.5) Return to 6;

  Otherwise STOP.
```

CARD+ implicitly partitions object data by deriving the distances from the relational data to a set of C implicit prototypes that summarize the data objects belonging to each cluster in the partition. Specifically, starting from the relation matrix $\mathbf{R}$, the following implicit distances are computed at each iteration step of the algorithm:

$$d_{ci} = (\mathbf{Rz}_c)_i - \mathbf{z}_c\mathbf{Rz}_c/2 \tag{12}$$

for all behavior vectors $i = 1, \ldots, n$ and for all implicit clusters $c = 1, \ldots, C$, where $\mathbf{z}_c$ is the membership vector for the cth cluster, defined on the basis of the fuzzy membership values z_{ci} that describe the degree of belongingness of the ith behavior vector in the cth cluster. Once the implicit distance values d_{ci} have been computed, the fuzzy membership values z_{ci} are updated to optimize the clustering criterion, resulting in a new fuzzy partition of behavior vectors. The process is iterated until the membership values stabilize.

Finally, a crisp assignment of behavior vectors to the identified clusters is performed in order to derive a prototype vector for each cluster, representing a user category. Precisely, each behavior vector is crisply assigned to the closest cluster, creating C clusters:

$$\chi_c = \left\{\mathbf{b}_i \in \mathbf{B} | d_{ci} < d_{ki} \forall c \neq k\right\}, \quad 1 \le c \le C. \tag{13}$$

Then, for each cluster χ_c a prototype vector $\mathbf{v}_c = (v_{c1}, v_{c2}, \ldots, v_{cm})$ is derived, where

$$v_{cj} = \frac{\sum_{\mathbf{b}_i \in \chi_c} b_{ij}}{|\chi_c|}, \quad j = 1, \ldots, m. \tag{14}$$

The values v_{cj} represent the significance (in terms of relevance degree) of a given page p_j to the cth user category.

However, this process fails to find the optimal number of clusters by deriving very overlapping clusters having low inter-distance values. Hence, to join the overlapping clusters into a single cluster, CARD+ includes a post-clustering process. This process enables the creation of well distinct clusters that correspond to categories reflecting the actual user preferences embedded in the available behavior matrix.

More precisely, the post-clustering process works as follows. For each pair of clusters, the inter-cluster distances are computed. These are defined as:

$$D_{ck} = \sum_{b_i \in \chi_c} \sum_{b_l \in \chi_k, i \neq l} \frac{||\mathbf{b}_i - \mathbf{b}_l||^2}{|\chi_c||\chi_k|} \tag{15}$$

where χ_c and χ_k are as defined in (13).

A high value of the inter-cluster distance means that the two clusters are separated, while a very low value means that clusters are very overlapping. To avoid redundant clusters, the following heuristic is applied. First, the average values of the inter-cluster distance between all cluster pairs is computed. Then, if the inter-cluster distance between two clusters drops below this average value, the two clusters are fused together into a single cluster that embraces behavior vectors of both clusters. The addition of this heuristic enables CARD+ to produce always the same partition of user behavior vectors, independently on the initial number of clusters $C_{\max}$.

The steps performed by CARD+ are summarized in Algorithm 1.

Once the clustering process has been completed, the CARD+ algorithm provides the following results:

- C cluster prototypes represented as vectors $\mathbf{v}_c = (v_{c1}, v_{c2}, \ldots, v_{cm})$ for $c = 1, \ldots, C$.
- A fuzzy partition matrix $\mathbf{M} = [m_{ic}]_{i=1,\ldots,n}^{c=1,\ldots,C}$ where m_{ic} represents the membership degree of the user behavior vector $\mathbf{b}_i$ to the cth cluster.

Each cluster prototype $\mathbf{v}_c = (v_{c1}, v_{c2}, \ldots, v_{cm})$ describes the typical navigational behavior of a group of users with similar interests about the most visited pages of the Web site.

Information obtained through categorization of user behaviors are exploited by NEWER in the successive step of knowledge discovery in order to construct the recommendation model useful to generate the dynamic suggestion of interesting links.

5.2. Discovery of the recommendation model

In this step, NEWER uses both behavior vectors and user categories identified by fuzzy clustering to extract a recommendation model that captures associations between behavior vectors and URLs to be recommended. Such model represents the knowledge base to be used by the online recommendation module of NEWER (see Fig. 1).

The discovery of association rules is performed through the learning of a neuro-fuzzy network, i.e. a neural network that encodes in its topology the structure of a Fuzzy Inference System (FIS) [13].

In NEWER, each rule in the FIS expresses a fuzzy relation between the components of a user behavior vector $\mathbf{b} = (b_1, b_2, \ldots, b_m)$ and the URLs to be recommended in the following form:

IF b_1 is A_{1k} AND . . . AND b_m is A_{mk}
THEN relevance of URL_1 is r_{1k} AND . . . AND relevance of URL_m is r_{mk}

for $k = 1, \ldots, K$ where K is the number of fuzzy rules, A_{jk}, $j = 1, \ldots, m$ are fuzzy sets with Gaussian membership functions defined over the input variables b_j and r_{jk}, $j = 1, \ldots, m$ are fuzzy singletons expressing the amount of recommendation (relevance degree) of the jth URL.

The main advantage of using a fuzzy knowledge base for the recommendation system is readability of the extracted knowledge. Actually, fuzzy rules can be easily understood by humans since they can be expressed in a linguistic fashion by labeling fuzzy sets A_{jk} with linguistic terms such as LOW, MEDIUM, and HIGH. Hence, a fuzzy rule in the web recommendation system can assume the following linguistic form:

IF (the degree of interest for URL_1 is LOW) AND . . . AND
(the degree of interest for URL_m is HIGH)
THEN (recommend URL_1 with relevance 0.3) AND . . . AND
(recommend URL_m with relevance 0.8)

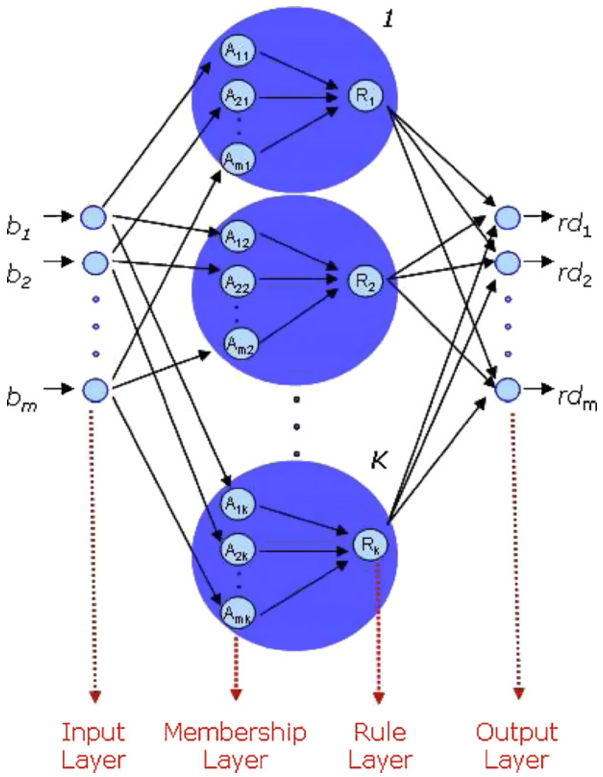


Fig. 2. The structure of the employed neural network.

To derive a fuzzy rule-based recommendation model, a four-layer feedforward neural network which reflects the fuzzy rule base in its parameters and topology is employed.

The structure of the neural network (depicted in Fig. 2) is composed of a set of units arranged into four layers which compute, respectively:

- the transferred input;
- the membership degree to fuzzy sets;
- the fulfillment degree for each fuzzy rule;
- the inferred outputs.

On the basis of the layer they belong to, units in the network have the following specifications:

(1) The first layer L_1 (the *input layer*) simply contains a number m units which transfer the crisp input values of the vector $\mathbf{b} = (b_1, b_2, \ldots, b_m)$. Hence, the units of this layer perform the trivial computation to produce the following output:

$$O_j^{(1)} = b_j, \quad j = 1, \ldots, m \tag{16}$$

(2) Units in the second layer L_2 (the *membership layer*) receive the degrees of interest for visited pages in a behavior vector $(b_1, b_2, \ldots, b_m)$ and evaluate the Gaussian membership functions representing fuzzy sets. In this layer, units are arranged in K groups, one for each fuzzy rule. The kth group contains m units corresponding to the fuzzy sets which define the premise part of the kth rule. In detail, each unit of the layer L_2 receives the interest degree for the jth page b_j, $j = 1, \ldots, m$ and computes its membership value to fuzzy set A_{jk} as follows:

$$O_{jk}^{(2)} = \exp\left(-\frac{(b_j - c_{jk})^2}{2a_{jk}^2}\right), \quad j = 1, \ldots, m, \quad k = 1, \ldots, K \tag{17}$$

where c_{jk} and a_{jk} are the center and the width of the Gaussian function, representing the adjustable parameters of units of L_2.

(3) The third layer L_3 (the *rule layer*) contains K units that compute the fulfillment degree of each rule. In this layer, no modifiable parameter is associated with the units. The output is derived by computing the rule activation strength, as follows:

$$O_j^{(3)} = \prod_{j=1}^{m} O_{jk}^{(2)}, \quad j = 1, \ldots, m \tag{18}$$

(4) The fourth layer L_4 (the *output layer*) provides the outputs of the network, i.e. the relevance values of the m URLs to be used for recommendation. Each relevance value is obtained by inference of rules, according to the following formula:

$$O_j^{(4)} = \frac{\sum_{k=1}^{K} O_k^{(3)} r_{jk}}{\sum_{k=1}^{K} O_k^{(3)}}, \quad j = 1, \ldots, m \tag{19}$$

Connections between layer L_3 and L_4 are weighted by the fuzzy singletons r_{jk} that represent a set of free parameters for the neuro-fuzzy network.

This neuro-fuzzy network is employed in order to derive the recommendation model expressed in the form of a set of fuzzy rules via a supervised learning.

The learning process of the neuro-fuzzy network assumes the availability of a dataset containing examples of associations between user behaviors and relevance of links to be suggested. To this aim, we firstly create a dataset that combines information about the available behavior data and user categories. Specifically, a set of n input–output samples describing the associations between each behavior vector and the relevance degrees for each page is built. The dataset is represented as follows:

$$\mathbf{T} = \left\langle (\mathbf{b}_i, \mathbf{rd}_i) \right\rangle_{i=1,\ldots,n} \tag{20}$$

where the input vector $\mathbf{b}_i$ represents the ith user behavior vector (row of the available behavior matrix), and the output vector $\mathbf{rd}_i$ expresses the amount of page recommendation for the ith user. To compute the values in $\mathbf{rd}_i$, we exploit information embedded in the user categories extracted through fuzzy clustering. Precisely, for each vector $\mathbf{b}_i$, we consider its membership to the user categories expressed by membership values $\{m_{ic}\}_{c=1,\ldots,C}$ in the partition matrix $\mathbf{M}$. Then, we identify the l top matching user categories $c_1, \ldots, c_l \in \{1, \ldots, C\}$ as those with the highest membership values[2]. The values in the output vector $\mathbf{rd}_i = (rd_{i1}, rd_{i2}, \ldots, rd_{im})$, $(i = 1, \ldots, n)$ are hence calculated as:

$$rd_{ij} = m_{ic_1}\mathbf{v}_{jc_1} + \cdots + m_{ic_l}\mathbf{v}_{jc_l}, \quad j = 1, \ldots, m \tag{21}$$

Starting from the constructed dataset, a training set and a test set are created by specifying a percentage of the total number n of samples as size of the training set (number of training samples). The test set is composed by the remaining samples.

Once the training set has been constructed, the neuro-fuzzy network can enter the learning phase to extract the knowledge embedded into the training set and represent it as a collection of fuzzy rules. Firstly, behavior data and extracted user categories are employed to initialize the structure and the involved parameters of the neuro-fuzzy network, deriving an initial fuzzy rule base. In particular, the number of fuzzy rules (and the number of fuzzy sets used to partition data) together with the parameters that define the premise and the consequence of each rule are established. A fuzzy rule is derived for each user category. Hence, calculation of the premise parameters of a rule depends on the center and the spread of the corresponding cluster. More precisely, from the kth cluster, a fuzzy rule is derived whose premise parameters are defined as follows:

- the vector $\mathbf{c}_k$ of centers of the Gaussian membership functions μ_{ik} coincides with the center $\mathbf{v}_k = (v_{1k}, v_{2k}, \ldots, v_{mk})$ of the kth cluster;
- the widths a_{jk} of the Gaussian functions are calculated by using the *first-nearest-neighbor* heuristic:

$$a_{jk} = \frac{||\mathbf{v}_k - \mathbf{v}_h||}{r} \tag{22}$$

where $\mathbf{v}_h$ is the cluster center nearest to $\mathbf{v}_k$ and r is an overlap parameter ranging in [1.0,2.0].

To derive the consequence parameters of each rule, all the input–output data are taken into account. Initial values of consequent parameters r_{jk} are obtained by defining a relationship between the membership values associated to clusters in the input space and the target values. Specifically, this relation is defined using the available target vectors $\mathbf{rd}_i = (rd_{i1}, rd_{i2}, \ldots, rd_{im})$, for all $i = 1, 2, \ldots, n$, and the information given by the membership values of the corresponding input vectors $\mathbf{b}_i$, $i = 1, 2, \ldots, n$ to the fuzzy clusters found in the input space.

Formally, the consequents values r_{jk} of the kth rule are obtained by weighting each of the data in the output domain by the degree of activation of the premise part of such a rule, in the following way:

$$r_{jk} = \frac{\sum_{i=1}^{n} \mu_k(\mathbf{b}_i) rd_{ij}}{\sum_{i=1}^{n} \mu_k(\mathbf{b}_i)}, \quad j = 1, 2, \ldots, m \tag{23}$$

with $\mu_k(\mathbf{b}_i)$ being the activation level of the premise part of the rule computed as in (18). These values can be computed by employing the values of the premise membership functions μ_{jk} that have been considered coinciding with the cluster center vectors as defined above.

Once the network structure has been established, the neural network enters in a learning phase to optimally adjust the premise and the consequent parameters of the fuzzy rules. Major details about the algorithm underlying the neuro-fuzzy learning can be retrieved in [3]. At the end of the learning process, a set of fuzzy rules is derived representing the recommendation model to be used by the online module to suggest interesting links.

6. Recommendation

The ultimate task of the NEWER system is the online recommendation of links to pages judged interesting for the current user of the Web site. Specifically, when a new user accesses the Web site, an on-line module matches his current partial session against the fuzzy rules currently available in the knowledge base and derives a vector of relevance degrees by means of a fuzzy inference process.

Formally, when a new user has access to the Web site, an active user's current session is created in the form of a vector $\mathbf{b}^0$. Each time the user requests a new page, the vector is updated. To maintain the active session, a sliding window is used to capture the most recent user's behavior. Thus the partial active session of the current user

[2] The number l of top matching session categories to be considered is experimentally established.

is represented as a vector $\mathbf{b}^{(0)} = \left(b_1^{(0)}, b_2^{(0)}, \ldots, b_m^{(0)}\right)$ where some values are equal to zero, corresponding to unexplored pages.

Based on the set of K rules generated through the learning of the neuro-fuzzy network, the recommendation module provides URL relevance degrees by means of the following fuzzy reasoning procedure:

(1) Calculate the matching degree of current session $\mathbf{b}^{(0)}$ to the kth rule, for $k = 1, \ldots, K$ by means of the product operator:

$$\mu_k(\mathbf{b}^{(0)}) = \prod_{j=1}^{m} \mu_{jk}(b_j^{(0)}) \tag{24}$$

(2) Calculate the relevance degree rd_j^0 for the jth URL, $j = 1, \ldots, m$ as:

$$rd_j^0 = \frac{\sum_{k=1}^{K} rd_{jk}\mu_k(\mathbf{b}^{(0)})}{\sum_{k=1}^{K} \mu_k\left(\mathbf{b}^{(0)}\right)} \tag{25}$$

This inference process provides the relevance degree for all the considered m pages, independently on the actual navigation of the current user. In order to perform dynamic link suggestion, the recommendation module firstly identifies URLs that have been not visited by the current user, i.e. all pages such that $b_j^0 = 0$. Then, among unexplored pages, only the first top-N pages having the highest relevance degree are recommended to the user. In practice, a list of links is dynamically included in the page currently visited by the user.

7. Experimental evaluation

To show the applicability and the potential of NEWER for the recommendation of interesting links, the system has been put into action by carrying out a set of experiments. The experimental sessions included: (i) an experimental simulation performed on a synthetic dataset; (ii) an application example executed on real-world data consisting in log files from a highly visited Web site.

In the following sections, for each simulation, the working process of NEWER is described in a step-by-step way and the results obtained after the execution of each step are shown.

7.1. Simulation on synthetic data

A preliminary experimental session was carried out on synthetic behavior data.

The synthetic dataset consists in a 434 × 10 behavior matrix $B = [b_{ij}]$, $i = 1, \ldots, 434$ and $j = 1, \ldots, 10$ where each component b_{ij} represents the interest degree of the ith user for the jth page. The interest degrees are expressed by real numbers in a value scale ranging from 0 to 1. The pages are labeled by $P_1, P_2, \ldots, P_{10}$.

Since these experiments were carried out on synthetic data, the preprocessing step was not performed. Consequently, this experimental session starts with the knowledge discovery phase that is aimed to the extraction of user categories by means of a fuzzy clustering process and the discovery of the recommendation model by the neuro-fuzzy strategy, as described in Section 5.

Knowledge discovery

Identification of user categories: Starting from the available behavior matrix, the CARD+ algorithm was applied in order to group behavior vectors into user categories. Several runs of the algorithm were performed by setting, in each trial, a different initial maximum number of clusters $C_{\max}$ with $C_{\max} = (5, 10, 15)$.

To evaluate the suitability of the fuzzy similarity measure, we also applied CARD+ equipped with the cosine measure. Moreover, to show the validity of CARD+, we compared it with the original CARD algorithm equipped with both similarity measures.

To evaluate the compactness of the derived partitions, two different indexes were calculated: the Dunn's index and the Davies–Bouldin index [7]. Large values of Dunn's index and low values for the Davies–Bouldin index correspond to good data partitions. Fig. 3 shows the values obtained for both validation indexes in the different runs. In correspondence of each trial, the figure indicates the final number of clusters extracted by CARD and CARD+. It can be observed that CARD+ with the use of the fuzzy similarity measure partitions the available data into 5 or 6 clusters. More precisely, 5 clusters were derived only in correspondence of $C_{\max} = 5$. CARD+ with the use of the cosine measure provided data partitions with the same final number of clusters $C = 5$. In this set of experiments, the validity indexes take the same values in all runs (the Dunn's index value was always equal to 0.54 and the value for the Davies–Bouldin index was 0.23). Hence, the employment of the cosine measure avoided CARD+ to identify one cluster, leading to the loss of a significant user category. Also, it can be observed the instability of the CARD algorithm with the use of both similarity measures, since it provided data partitions with different final number of clusters in each trial. Comparing the values obtained for the validity indexes in the different runs, the partition obtained by CARD+ with the use of the fuzzy similarity measure turns out be the best partition of the available dataset. This conclusion is supported by different observations. Firstly, although the CARD+ algorithm with cosine measure revealed to be a robust approach to categorize user behavior vectors, it has partitioned data into only $C = 5$ clusters. Instead, CARD+ with the fuzzy measure grouped the available data into 6 different clusters, thus discovering a further cluster encoding the interests of a specific user category. The CARD algorithm revealed to be not very stable, either with the employment of cosine measure or the fuzzy similarity measure.

The information about the user categories derived by CARD+ with the use of the fuzzy similarity measure are summarized in Table 1. For each user category (labeled with numbers 1, 2, . . ., 6) the pages with the highest degree of interest are indicated. The

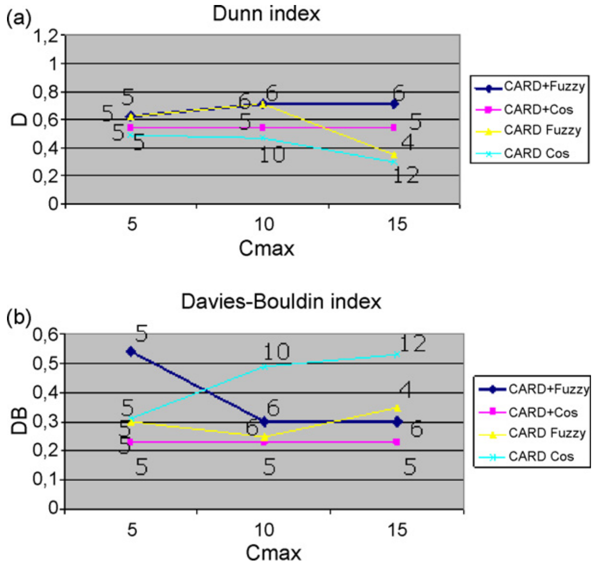


Fig. 3. The Dunn's index (a) and the Davies–Bouldin index (b) obtained by CARD+ and CARD on synthetic data.

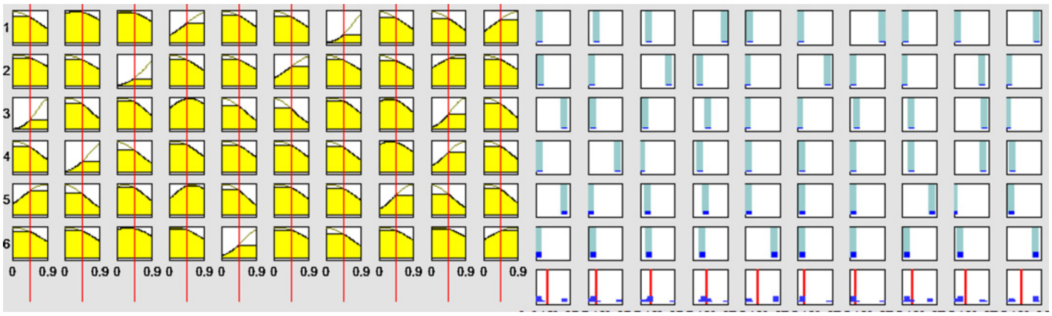


Fig. 4. The final recommendation model derived from synthetic data.

last column of the table reports the common access pages characterizing the corresponding user category. It can be noted that some pages (e.g. P_1 and P_9) are included in more than one user category, showing how different categories of users may exhibit common interests. This also confirms the importance of using fuzzy clustering techniques which permit to produce overlapping clusters.

Discovery of the recommendation model: Successively, the information about the identified user categories in combination with behavior vector data were used to create the dataset necessary for the generation of the recommendation model. A dataset of 434 input–output samples was composed in the way described in Section 5.2. Each sample includes 20 components (10 corresponding to the pages of each behavior vector and the remaining 10 calculated as in formula (21)).

Once the dataset was composed, an initial recommendation model expressed in the form of fuzzy rule base was derived and embedded in a neuro-fuzzy network with 10 inputs (corresponding to the pages of the behavior vectors) and 10 outputs (corresponding to the relevance values of the Web pages). The obtained rule base is composed of 6 fuzzy rules, one for each extracted user category. Then, the neuro-fuzzy learning was applied to derive a final rule base containing the associations between the user behavior vectors and the relevancies of each URL. In order to obtain more reliable results, a 10-fold cross-validation procedure was performed. The dataset was partitioned in 10 subsets and, in each run, one subset was used as test set to evaluate the fuzzy rule base obtained by neuro-fuzzy learning run over the union of the remaining 9 subsets (considered as training set). In each learning run, the network was trained until the error on the training set dropped below 0.01 or the total number of executed epochs achieves 1000. At the end of the 10-fold cross-validation procedure, we chose as final recommendation model the fuzzy rule base corresponding to the trial providing the lowest error on the test set.

The derived fuzzy rule base (final recommendation model) is depicted in Fig. 4. It can be seen that for each input variable, only two distinct fuzzy sets are identified. This enables an easy interpretation of the fuzzy rules, that can be expressed in a linguistic fashion, by associating a linguistic label (LOW or HIGH) to each fuzzy set. As an example, Fig. 5 describes the first fuzzy rule of the final recommendation model expressed in the linguistic form.

Table 1
User categories identified on synthetical data.

User category	Interest degree	Common access pages
1	$P_4 = 0.88, P_7 = 0.85, P_{10} = 0.82$	$\{P_4, P_7, P_{10}\}$
2	$P_3 = 0.87, P_6 = 0.84, P_9 = 0.75$	$\{P_3, P_6, P_9\}$
3	$P_1 = 0.82, P_9 = 0.80$	$\{P_1, P_9\}$
4	$P_2 = 0.86, P_9 = 0.80$	$\{P_2, P_9\}$
5	$P_1 = 0.85, P_8 = 0.81$	$\{P_1, P_8\}$
6	$P_5 = 0.88, P_{10} = 0.84$	$\{P_5, P_{10}\}$

Recommendation

Finally, the generated fuzzy rule base was used to infer the relevance degree of each URL for the current user. On the basis of the relevance degrees obtained through the fuzzy inference process, it was possible to suggest a list of links to unexplored pages retained the most interesting to the user. The list of recommended pages was derived by following the procedure based on the fuzzy inference reasoning described in section 6.

7.2. Simulation on real-world data

A further experimental simulation concerned the test of NEWER on real-world data consisting in the access log files from an Italian Web site of the Japanese movie Dragon Ball (http://www.dragonballgt.it). This Web site was considered for its high number of daily accesses (thousands of visits each day), especially from younger people. In particular, log data collected during a period of 12 h (from 10:00 a.m. to 22:00 p.m.) were considered in this experimental simulation.

Log file preprocessing

Firstly, all the 12,300 entries of the log file were mapped into a relational database, including entries that recorded server request failures, authentication failures, or also entries referring

IF (interest degree for P_1 is LOW) AND
(interest degree for P_2 is LOW) AND
(interest degree for P_3 is LOW) AND
(interest degree for P_4 is HIGH) AND
(interest degree for P_5 is LOW) AND
(interest degree for P_6 is LOW) AND
(interest degree for P_7 is HIGH) AND
(interest degree for P_8 is LOW) AND
(interest degree for P_9 is LOW) AND
(interest degree for P_{10} is HIGH)
THEN (recommend P_1 with relevance 0.07) AND
(recommend P_2 with relevance 0.04) AND
(recommend P_3 with relevance 0.08) AND
(recommend P_4 with relevance 0.35) AND
(recommend P_5 with relevance 0.06) AND
(recommend P_6 with relevance 0.07) AND
(recommend P_7 with relevance 0.40) AND
(recommend P_8 with relevance 0.06) AND
(recommend P_9 with relevance 0.08) AND
(recommend P_{10} with relevance 0.67)

Fig. 5. Linguistic expression of the first recommendation rule.

to accesses to different kinds of objects, such as images, videos, etc. Then, data cleaning was carried out in order to remove noisy and all entries related to kinds of requests that were not considered useful. At the end of data cleaning, the number of log entries was reduced to 5057, representing about the 41% of the initial number of log entries. The high number of deleted entries is explained by the fact that in the considered Web site there is a consistent number of images and videos. Thus, when users request a page, entries for the graphical objects contained in the requested pages are implicitly stored in log files. Indeed, since the site users are often represented by very young people, the number of failed requests results to be high. Table 2 contains the summary of the information extracted by the data cleaning activity.

In the next activity of data structuration, user sessions were identified by grouping the requests originating from the same IP address within a prefixed time period (as described in Section 4). In these experiments, we fixed the maximum value $\Delta t_{\max}$ for the elapsed time between two consecutive requests to 30 min and the minimum value $\Delta t_{\min}$ to 3 s. After data structuration, the 5057 retained entries were structured into a final number of 1480 user sessions and a total number of 870 distinct URLs accessed in these sessions was identified.

Once user sessions were identified, we performed data filtering. Support-based data filtering was used to eliminate requests for URLs having a number of accesses less than 10% of the maximum number of accesses, leading to only 159 distinct URLs and 840 sessions. Also, URLs appearing in more than 80% of sessions (including the site entry page) were filtered out, leaving 42 final URLs and 575 sessions. The values adopted for all the thresholds involved in data preprocessing step were chosen by committing to previous applications of log file preprocessing on access data from other sites. Finally, data filtering eliminated short sessions, leaving only sessions with at least 8 distinct requests. A final number of 200 sessions was obtained. The 42 pages in the Web site were labeled with a number (see Table 3) to facilitate the analysis of results, by specifying the content of the retained Web pages.

Table 2
A summary of the entries removed by the data cleaning activity.

Request type	Cleaned requests	% Initial size
.jpg	1168	9.5
.ico	0	0
.jpeg	923	7.5
.gif	307	2.5
.mp3	431	3.5
.wav	246	2
.png	307	2.5
.bmp	0	0
.mpeg	615	5
.css	246	2
.js	185	1.5
.wmv	0	0
.mid	307	2.5
.avi	0	0
.tif	0	0
.swf	0	0
.ram	0	0
.rm	123	1
.pdf	369	3
.txt	0	0
.doc	0	0
POST	246	2
OPTION	0	0
HEAD	0	0
PROPFIND	185	1.5
Robots	246	2
Failed request	615	5
Directory	246	2
Corrupt line	369	3
Code ≠ 200	123	1

Table 3
Description of the retained pages in the Web site.

Pages	Content
1, . . ., 8	Pictures of characters
9, . . ., 13	Various kind of pictures related to the movie
14, . . ., 18	General information about the main character
19, 26, 27	Matches
20, 21, 36	Services (registration, login, etc.)
22, 23, 24, 25, 28, . . ., 31	General information about the movie
32, . . ., 37	Entertainment (games, videos, etc.)
38, . . ., 42	Description of characters

Finally, we computed the interest degrees of each user for each visited page in the way described in Section 4, by using formula (3).

As a result, we obtained a 200 × 42 behavior matrix containing the interest degrees of each user for each page.

Knowledge discovery

Identification of user categories. Starting from the created behavior matrix, CARD+ was applied by using the fuzzy similarity measure. Several trials of CARD+ were performed by setting a different initial number of clusters $C_{\max} = (5, 10, 15)$. As before, the values of the Dunn's index and the Davies–Bouldin index were computed in order to establish the goodness of the derived categories of behavior vectors. Once again, we found that CARD+ with the use of the fuzzy similarity measure provided always the same final number of clusters $C = 5$, independently from the initial number of clusters $C_{\max}$. The validity indexes took the same values in all runs. In particular, the Dunn's index value was always equal to 1.35 and the value for the Davies–Bouldin index was 0.13. As a consequence, the CARD+ algorithm equipped with the fuzzy similarity measure resulted to be quite stable, by partitioning the available behavior data into 5 clusters corresponding to the identified user categories.

Analogously to the simulation on synthetic data, we also identified user categories by applying CARD equipped with the cosine and fuzzy similarity measures. In Fig. 6, the obtained values of the Dunn's index and the Davies–Bouldin index are shown and the final number of clusters extracted by CARD+ and CARD corresponding to each trial is also indicated. As it can be observed, CARD+ with the use of the cosine measure derived partitions which categorized

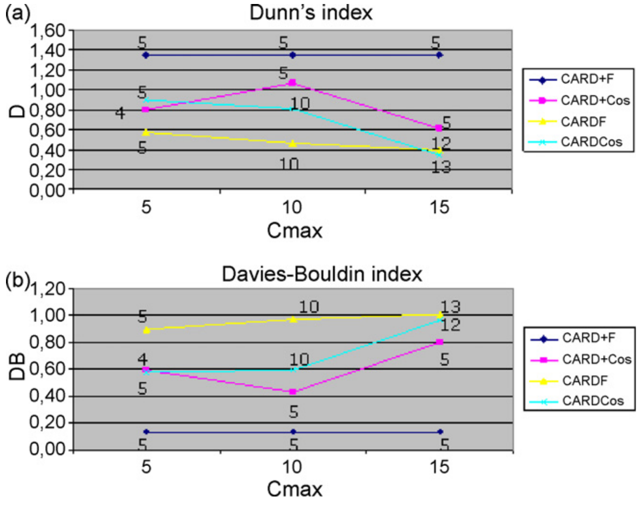


Fig. 6. The Dunn's index (a) and the Davies–Bouldin index (b) obtained by CARD+ and CARD on real world data.

Table 4
User categories identified on real-world data.

User category	Relevant pages (interest degrees)
1	P_1(55), P_2(63), P_3(54), P_5(52), P_7(48), P_8(43), P_{14}(66), P_{28}(56), P_{29}(52), P_{30}(37)
2	P_1(72), P_2(59), P_3(95), P_6(65), P_7(57), P_{10}(74), P_{11}(66), P_{13}(66)
3	P_1(50), P_2(50), P_3(45), P_4(46), P_5(42), P_6(42), P_8(34), P_9(37), P_{12}(40), P_{15}(41), P_{16}(41), P_{17}(38), P_{18}(37), P_{19}(36)
4	P_2(49), P_{10}(47), P_{11}(38), P_{12}(36), P_{14}(27), P_{31}(36), P_{32}(29), P_{33}(39), P_{34}(36), P_{35}(26), P_{36}(20), P_{37}(37), P_{38}(29), P_{39}(30), P_{40}(34), P_{41}(28), P_{42}(24)
5	P_4(70), P_5(65), P_{20}(64), P_{21}(62), P_{22}(54), P_{23}(63), P_{24}(54), P_{25}(41), P_{26}(47), P_{27}(47)

data into 4 or 5 clusters, resulting less stable than CARD+ equipped with the fuzzy similarity measure.

Once again, the CARD algorithm showed an instable behavior, by providing data partitions with a different final number of clusters in each trial. The CARD algorithm had an analogous behavior when equipped with the fuzzy similarity measure.

Analyzing the results obtained by the different runs, we can draw some conclusions. Firstly, a good partition categorizes the available data into 5 final clusters. The CARD algorithm did not reveal to be very stable, with the use of both the similarity measures. Finally, by comparing the values of the obtained validity indexes, we observed that the best values correspond to the partition derived by CARD+ with the employment of the fuzzy similarity measure.

Based on these observations, we can conclude that CARD+ equipped with the fuzzy similarity measure was able to derive the best partition in terms of compactness. As a consequence, we choose to use the partition derived by this approach for the successive step of recommendation model discovery.

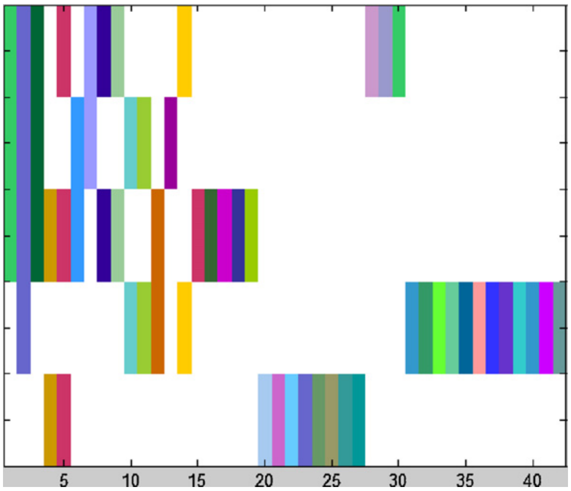
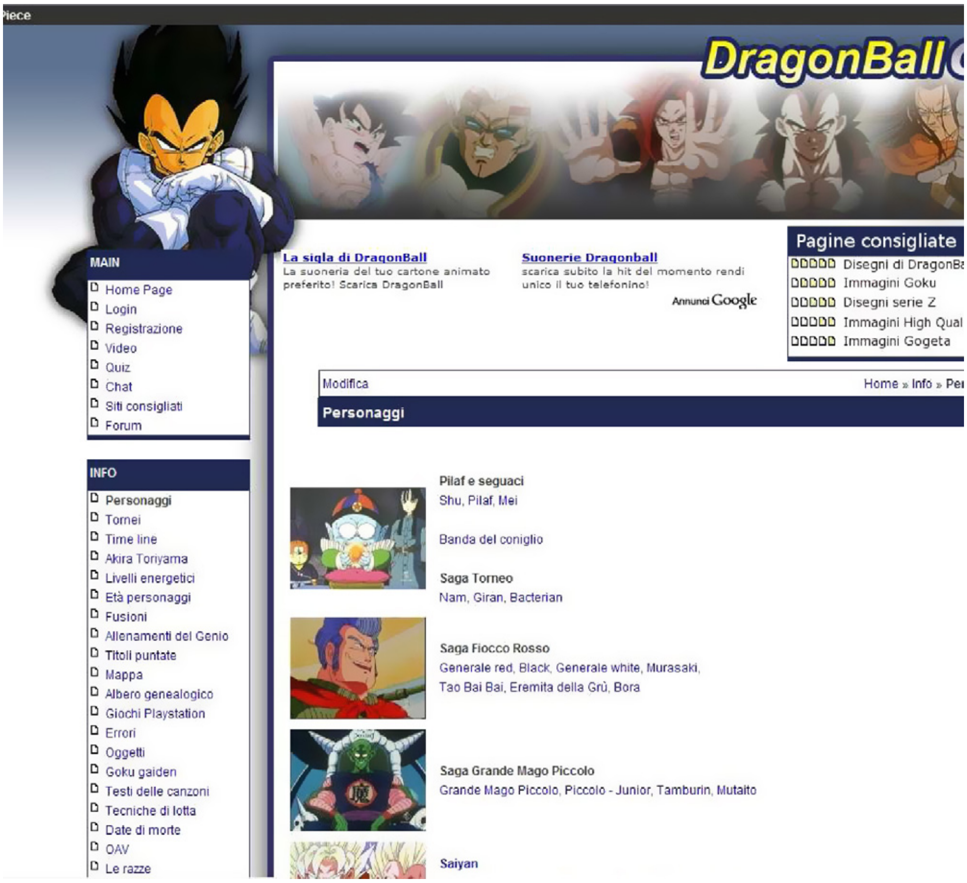


Fig. 7. Graphical representation of the user categories identified on real-world data.

User categories corresponding to the chosen partition are summarized in Table 4. For each user category (labeled with numbers 1, 2, . . ., 5) the pages and the relevance degree are indicated. To provide a more immediate visualization of the common access pages in the extracted user categories, Fig. 7 illustrates a graphical representation of these. In this figure, each row represents a user category and each column represents a page. Different colors are used for different pages; for pages which are not visited the white color is



La sigla di DragonBall
La suoneria del tuo cartone animato preferito! Scarica DragonBall

Suonerie Dragonball
scarica subito la hit del momento rendi unico il tuo telefonino!

Pagine consigliate
□□□□□ Disegni di DragonB
□□□□□ Immagini Goku
□□□□□ Disegni serie Z
□□□□□ Immagini High Qual
□□□□□ Immagini Gogeta

Personaggi

Pilaf e seguaci
Shu, Pilaf, Mei

Banda del coniglio

Saga Torneo
Nam, Giran, Bacterian

Saga Fiocco Rosso
Generale red, Black, Generale white, Murasaki, Tao Bai Bai, Eremita della Grù, Bora

Saga Grande Mago Piccolo
Grande Mago Piccolo, Piccolo - Junior, Tamburin, Mutaito

Saiyan

Fig. 8. An example of Web pages displaying a generated recommendation list.

used. It can be noted that some pages (e.g. P_1, P_2, P_3, P_{10}, P_{11}, and P_{12}) are included in more than one user category, showing how different categories of users may exhibit common interests.

We can give an interpretation of the identified user categories, by individuating the interests of users belonging to each of these. The interpretation is indicated in the following.

- *Category 1*. Users in this category are mainly interested on information about the movie characters.
- *Category 2*. Users in this category are interested in the history of the movie and in pictures of movie and characters.
- *Category 3*. These users are mostly interested to the main character of the movie.
- *Category 4*. These users prefer pages that link to entertainment objects (games and video).
- *Category 5*. Users in this category prefer pages containing general information about the movie.

Discovery of the recommendation model. Once the user categories were identified, we created the dataset necessary to discover the recommendation model in the way described in Section 5.2. The created dataset was composed of 200 input–output samples, where each sample included 84 components (42 corresponding to the pages of each behavior vector and the remaining 42 calculated by employing formula 21).

Once the dataset was composed, an initial recommendation model expressed in the form of 5 fuzzy rules was set. Successively, the neuro-fuzzy strategy was applied to learn the recommendation model containing the associations between the user behavior vectors and the relevance degrees of each URL. A neural network having 42 inputs (corresponding to the URL of the user behavior vectors) and 42 outputs (corresponding to the relevance values of the Web pages) was considered. In order to evaluate the quality of the recommendation model, a 10-fold cross-validation procedure was performed. Among the 10 created models, we chose as final recommendation model, that model having the lowest error on the test set. The derived recommendation model is represented by a fuzzy rule base composed of 5 rules. Each rule is characterized by 42 input variables and 42 output.

Recommendation

Successively, the discovered recommendation model was used to infer the relevancies of each URL for a current user through the fuzzy inference procedure described in Section 6. Hence, the pages not yet visited by the current user were ordered on the basis of the corresponding relevance degrees and the 5 pages with the highest values were recommended. The generated list of recommended links is visualized in a section of the Web page currently visited by the user.

Fig. 8 shows an example of personalized page of the considered Web site displaying the list of links suggested by our recommendation system to a connected user.

8. Comparison with other recommendation approaches

The accuracy of recommendations generated by NEWER was evaluated by means of *Precision*, *Recall* and $F1$ measures. In order to calculate the values for these accuracy metrics, we considered the training set and the test set corresponding to the trial which has derived the final recommendation model with the lowest error on the test set. Then, Web pages visited in each of the behavior vectors included in the test set were divided randomly into two sub-sets, namely input set and measurement set. In particular, each input set was treated as an active user behavior (i.e. the behavior vector of a current user) and it was given in input to the recommendation process to determine the list of recommended pages or recommended set. Once this procedure was completed, the mean values of *Precision*, *Recall* and $F1$ measures were calculated.

To show the effectiveness of the NEWER system, a comparison with other approaches was carried out, both on synthetic and real world data.

For the comparison, three different recommendation approaches proposed in literature have been used:

- The *Nearest Profile* based recommendation approach (NP). In our comparison experiments, the distance between the current behavior vector and the identified user categories is evaluated on the basis of the cosine similarity measure.
- The recommendation approach based on the *fuzzy approximate reasoning* (FAR). This approach has been proposed by Nasraoui and Petenes [12]. It is based on a fuzzy inference procedure to infer the recommendation list. For this approach two variants are implemented on the basis of the type of operators used for the t-norm/intersection and t-conorm/union in the composition of the recommendation inference procedure: max–min (FARSP) and sum-product (FARMM).
- The *K-Nearest Neighbors* (KNN). In our experiments, the values of K and N are respectively fixed to 10 and 5.

As concerns the first two recommendation approaches, we used the user categories extracted by CARD+ equipped with the fuzzy similarity measure as user behavior models to be exploited by the recommendation approaches.

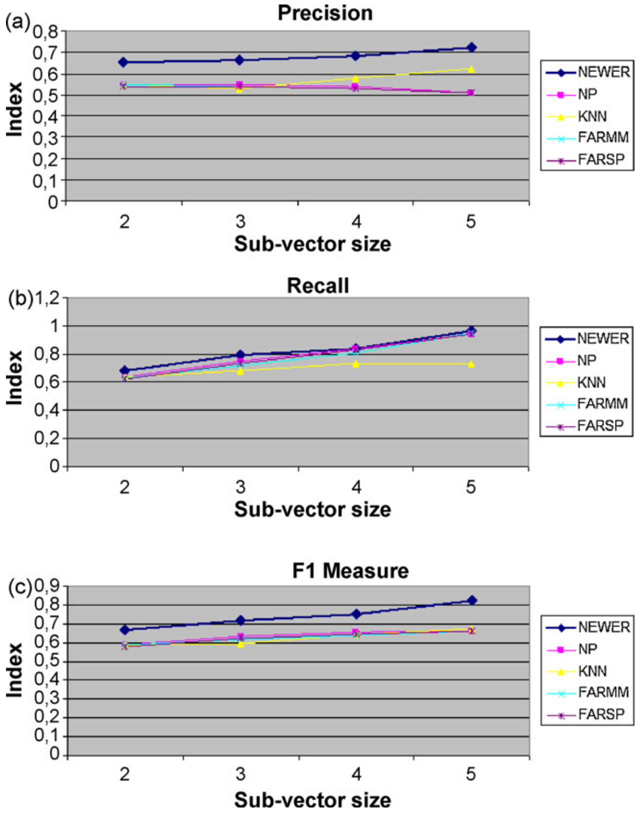


Fig. 9. Comparison of average precision (a), recall (b), F1 (c) per sub-vector size between the proposed personalization approach with the other recommendation approaches on synthetic data.

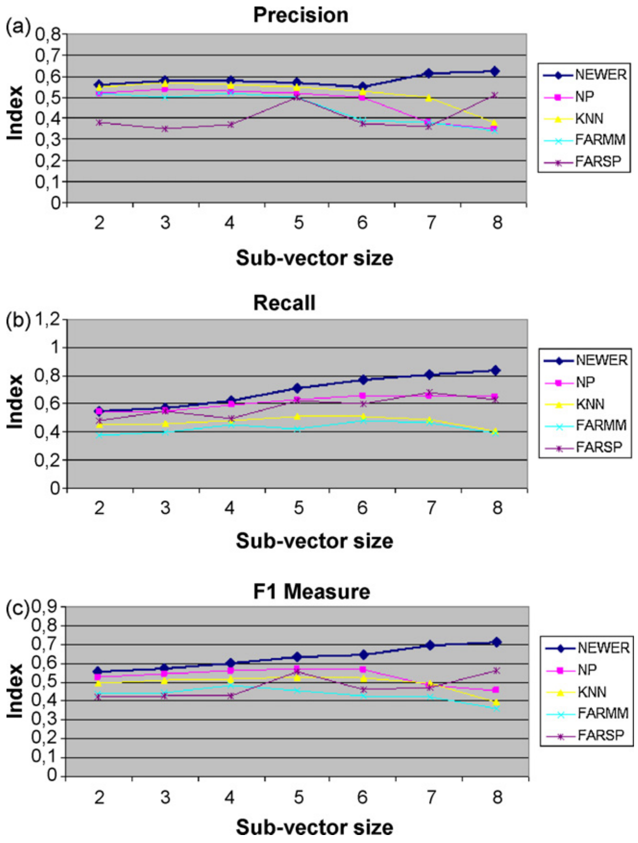


Fig. 10. Comparison of average precision (a), recall (b), F1 (c) per sub-vector size between the proposed personalization approach with the other recommendation approaches on real-world data.

Fig. 9 shows the comparative values of *Precision*, *Recall* and *F*1 on synthetic data. It can be observed that recommendations generated by NEWER are better than those obtained with the other approaches, especially in correspondence to higher sub-vector sizes. The better performance for longer sub-vector is due to the fact that longer vectors match more likely with different user categories. In this situation, a neuro-fuzzy approach is expected to be more effective, because it enables a user to belong to several categories with different membership degrees. Similar considerations can be made by comparing the accuracy values obtained by all these approaches on the real-world data, illustrated in Fig. 10.

9. Future research and conclusions

In this paper, we present NEWER, a Web personalization system designed to dynamically suggest interesting links to the current users according to their interests. In such a system, useful knowledge about usage access patterns is mined through the application of techniques underlying the WUM methodology. The examples given throughout this paper highlight that the NEWER system can be effective for recommendation, leading to a quality of the generated recommendations comparable and often significantly better than those of the other approaches of literature employed for the comparison.

On the overall, the reported results indicate that our proposed approach provides a valid tool to automatically extract fuzzy models useful for the generation of recommendations.

However, it is our belief that more extensive experimental comparisons should be carried out to produce a more definitive sentence on the superiority of our personalization system over another one.

Furthermore, our approach should be tested in many other real Web sites before being claimed as a consolidate tool.

The obtained promising results encourage the application of NEWER to a wider range of real-world data. All these facets are the subject of our on-going research projects.

References

[1] M. Albanese, A. Picariello, C. Sansone, L. Sansone, A web personalization system based on web usage mining techniques, in: Proceedings of the WWW2004, 2004.

[2] R. Baraglia, P. Palmerini, Suggest: a web usage mining system, in: Proceedings of the International Conference on Information Technology: Coding and Computing, 2002.

[3] G. Castellano, C. Castiello, A.M. Fanelli, C. Mencar, Knowledge discovering by a neuro-fuzzy modelling framework, Fuzzy Sets and Systems 149 (2005) 187–207.

[4] G. Castellano, A.M. Fanelli, C. Mencar, M.A. Torsello, Similarity-based fuzzy clustering for user profiling, in: Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, Silicon Valley, CA, 2007.

[5] G. Castellano, A.M. Fanelli, M.A. Torsello, Web user profiling using relational fuzzy clustering, in: P.Y. Cao, H. Cheng, D. Hung, C. Kahraman, C.W. Ngo, Y. Ohsawa, M.G. Romay, M.C. Su, A. Vasilakos, D. Wang, P. Wang (Eds.), Information Sciences 2007, World Scientific Publishing Co. Pte. Ltd., 2007, pp. 1433–1439.

[6] R. Cooley, B. Mobasher, J. Srivastava, Data preparation for mining worldwideweb browsing patterns, Knowledge and Information Systems 1 (1) (1999) 32–55.

[7] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster validity methods: Part ii, SIGMOD Record September (2002).

[8] X. Jin, B. Mobasher, Y. Zhou, A web recommendation system based on maximum entropy, in: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC05), 2005, pp. 213–218.

[9] B. Mobasher, R. Cooley, J. Srivastava, Creating adaptive web sites through usage-based clustering of urls, in: Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99), 1999.

[10] B. Mobasher, R. Cooley, J. Srivastava, Automatic personalization based on web usage mining, Communications of the ACM 43 (8) (2000) 142–151.

[11] O. Nasraoui, H. Frigui, Extracting web user profiles using relational competitive fuzzy clustering, International Journal on Artificial Intelligence Tools 9 (4) (2000) 509–526.

[12] O. Nasraoui, C. Petenes, Combining web usage mining and fuzzy inference for website personalization, in: Proceedings of WEBKDD 2003: Web Mining as Premise to Effective Web Applications, 2003, pp. 37–46.

[13] D. Nauck, F. Klawonn, R. Kruse, Foundations of NeuroFuzzy Systems, Wiley, 1997.

[14] D.S. Ngu, X. Wu, Sitehelper: a localized agent that helps incremental exploration of the world wide web, Computer Networks and ISDN Systems: The International Journal of Computer and Telecommunications Networking 29 (8) (1997) 1249–1255.

[15] G. Paliouras, C. Papatheodoru, V. Karkaletsis, P. Tzitziras, C.D. Spyropoulos, Large-scale mining of usage data on web sites, in: Proceedings of the AAAI Spring Symposium on Adaptive User Interface, Stanford, CA, 2000, pp. 92–97.

[16] J. Pei, J. Han, B. Motazavi-Asl, H. Zhu, Mining access patterns efficiently from web logs, in: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000, pp. 396–407.

[17] D. Pierrakos, G. Paliouras, C. Papatheodorou, C.D. Spyropoulos, A web usage mining tool for personalization, in: Proceedings of the Panhellenic Conference on Human Computer Interaction, 2001.

[18] C.E. Russell, S. Yuhui, Computational Intelligence, Morgan Kaufmann Publishers, 2007.

[19] S. Santini, R. Jain, Similarity measures, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (September 9) (1999).

[20] T.W. Yan, M. Jacobsen, H. Garcia-Molina, D. Umeshwar, From user access patterns to dynamic hypertext linking, in: Proceedings of the Fifth International World Wide Web Conference, 1996.

[21] L. Zhizhen, S. Pengfei, Similarity measures on intuitionistic fuzzy sets, Pattern Recognition Letter 24 (15) (2003) 2687–2693.