Complete this Puzzle: A Connectionist Approach to Accurate Web Recommendations based on a Committee of Predictors

Olfa Nasraoui[1] and Mrudula Pavuluri[2]

[1] Dept. of Computer Science and Engineering
Speed Scientific School, University of Louisville
Louisville, KY 40292

[2] Dept. of Electrical and Computer Engineering
The University of Memphis
Memphis, TN 38152-3180

Abstract. We present a *Context Ultra-Sensitive Approach based on two-step Recommender systems (CUSA-2-step-Rec)*. Our approach relies on a committee of profile-specific neural networks. This approach provides recommendations that are accurate and fast to train because only the URLs relevant to a specific profile are used to define the architecture of each network. Similar to the task of completing the missing pieces of a puzzle, each neural network is trained to predict the missing URLs of several complete ground-truth sessions from a given profile, given as input several incomplete subsessions. We compare the proposed approach with collaborative filtering showing that our approach achieves higher coverage and precision while being faster, and requiring lower main memory at recommendation time. While most recommenders are inherently context sensitive, our approach is context *ultra*-sensitive because a *different* recommendation model is designed for *each profile* separately.

Keywords: personalization, recommender systems, collaborative filtering, web usage mining, neural networks

1 Introduction

The Web information age has brought a dramatic increase in the sheer amount of information (content), the accessibility to this information (usage), as well as the intricate complexities governing the relationships within this information (structure). Hence, not surprisingly, information overload, when searching and browsing the WWW, has become the plague du jour. One of the most promising and potent remedies against this plague comes in the form of *personalization*. Personalization aims to customize the interactions on a website depending on the user's explicit and/or implicit interests and desires. The move from *traditional* physical stores of products or information (such as grocery stores or libraries) to virtual stores of products or information (such as *e-commerce sites* and *digital libraries*) has practically eliminated physical constraints traditionally limiting the number and variety of products in a typical inventory. Unfortunately, the move from the physical to the virtual space has drastically limited the traditional three dimensional layout of products for which access is further facilitated thanks to the sales representative or librarian who *know* their *products and* their *customers*, to a dismal *planar* interface *without* the sales representative or librarian. As a result, the customers are drowned by the huge number of options, most of which they may never even get to know. Hence, in both the e-commerce sector and digital libraries, Web personalization has become more of a necessity than an option. One of the most successful examples of personalization comes in the form of *recommender systems*. Several approaches to automatically generate Web recommendations based on user's Web navigation patterns or ratings exist. Some involve learning a usage model from Web access data or from user ratings. For example, lazy modeling is used in collaborative filtering which simply stores all users' information and then relies on *K-Nearest-Neighbors* (*KNN*) to provide

recommendations from the previous history of similar users. *Frequent itemsets*, session *clusters*, or user *profiles* can also form a user model obtained using data mining. Pazzani and Billsus [3] presented a collaborative filtering approach based on users' ratings of web pages, and Naives Bayes as the prediction tool. Mobasher et al. [1] use pre-discovered association rules and an efficient data structure to provide recommendations based on web navigation patterns. Among the most popular methods, the ones based on collaborative filtering and the ones based on fixed support association rule discovery may be the most difficult and expensive to use. This is because, for the case of high-dimensional and extremely sparse Web data, it is difficult to set suitable support and confidence thresholds to yield reliable and complete web usage patterns. Similarly, collaborative models may struggle with sparse data, and do not scale well to the number of users.

In this paper, we investigate several single-step and two-step recommender systems. The *Context Sensitive Approaches based on single-step Recommender systems (CSA-1-step-Rec)* simply predict the URLs that are part of the nearest estimated profile as recommendations. The nearest profile prediction model simply bases its recommendations on the closest profile. The *Context Ultra-Sensitive Approaches based on two-step Recommender systems (CUSA-2-step-Rec)* first maps a user session to one of the pre-discovered profiles, and then uses one of several profile-specific URL-predictor neural networks (such as Multilayer Perceptron or Hopfield Autoassociative memory networks) in the second step to provide the final recommendations. Based on this classification, a different recommendation model is designed for each profile separately. Each neural network is trained to complete the missing URLs of several complete ground-truth sessions from a given profile, given as input several incomplete subsessions. This learning is analogous to completing some missing parts of a puzzle. The two-step recommendation method not only handles *overlap* in user interests, but also can mend the effects of some types of misclassifications in the first nearest profile assignment step, and even mend the effect of a coarse profile dichotomy due to the profile discovery stage.

The rest of the paper is organized as follows. In Section 2, we present an overview of profile discovery using Web usage mining. In Section 3, we present the single-step profile prediction based recommendation process, and the two-step recommender system based on a committee of profile-specific URL-predictor neural networks. In Section 4, we present an empirical evaluation of the recommendation strategies on real web usage data, and finally, in Section 5, we present our conclusions.

2 Profile Discovery based on Web Usage Mining

Our approach is based on first extracting user *profiles* or *ratings* using a method, such as Web usage mining. In this case, the profile discovery can be executed *offline* by mining user access log files using the following steps:

(1) Preprocess log file to extract user *sessions*,
(2) *Categorize* sessions by *clustering*,
(3) Summarize the session categories in terms of *user profiles*,

After automatically grouping sessions into different clusters, we summarize the session categories in terms of *user profile vectors*, $\boldsymbol{p}_i$**:** The k^{th} component/weight of this vector ($\boldsymbol{p}_{ik}$) captures the *relevance* of URL_k in the i^{th} profile, as estimated by the conditional probability that URL_k is accessed in a session belonging to the i^{th} cluster.

3 Description of the Single-Step and Two-Step Recommendation Strategy Options

Let $U = \{\text{url}_1, \text{url}_2, \ldots, \text{url}_{N_U}\}$ be a set of N_U urls on a given web site visited in web user sessions $\boldsymbol{s}_j$, $j = 1, ..., N_s$, as defined in (1). Let $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{N_P}\}$ be the set of N_P Web user profiles computed by the profile discovery engine. Each profile consists of a set of URLs associated with their relevance weights in that profile. The problem of recommendation can be stated as follows. Given a current Web user session vector, $\boldsymbol{s}_j = [\text{s}_{j1}, \text{s}_{j2}, \ldots, \text{s}_{jN_U}]$, predict the set of URLs that are most relevant according to the user's interest, and recommend them to the user, usually as a set of Hypertext *links* dynamically appended to the contents of the Web document returned in response to the most recent Web query. It may be useful to associate the k^{th} recommended URL with a corresponding URL relevance *score*, r_{jk}. Hence it is practical to denote the recommendations for current Web user session, $\boldsymbol{s}_j$, by a vector $\boldsymbol{r}_j = [\text{r}_{j1}, \text{r}_{j2}, \ldots, \text{r}_{jN_U}]$. In this study, we limit the scores to be binary.

3.1 Context Sensitive Approach Based on Single-Step Profile Prediction Recommender System (*CSA-1-step-Rec*)

3.1.1 Single-Step Nearest-Profile Prediction Based Recommender System

The simplest and most rudimentary approach to profile based Web recommendation is to simply determine the most similar profile to the current session, and to recommend the URLs in this profile, together with their URL relevance weights as URL recommendation scores.

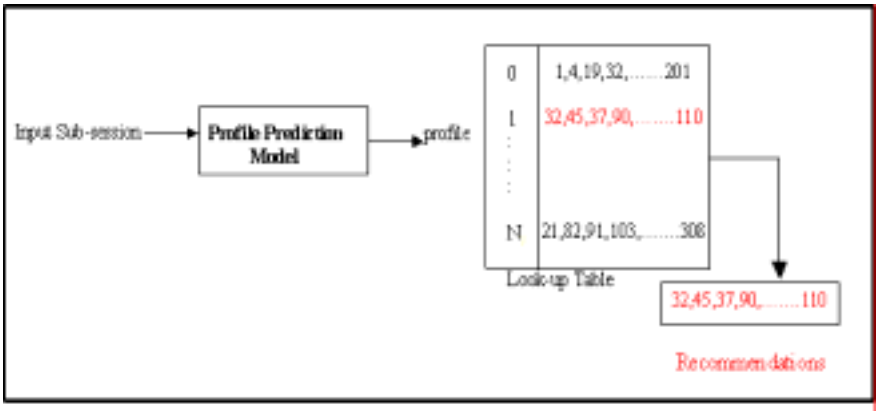


Fig. 1. Context-Sensitive Approach based on single-step profile prediction based Recommender System *(CSA-1-step-Rec)*. The Profile Prediction Model can be a Nearest-Profile classifier or any of the models shown in Figs 2 or 3.

Figure 1 shows the structure of such a recommendation system, where the profile prediction model simply consists of a nearest-profile estimator based on computing a session to profile similarity, and selecting the profile with highest similarity as the predicted profile.

The similarity score between an input session, $\boldsymbol{s}$, and the i^{th} profile, $\boldsymbol{p}_i$, can be computed using the cosine similarity as follows,

$$S_{si}^{cosine} = \frac{\sum_{k=1}^{N_U} p_{ik} s_k}{\sqrt{\sum_{k=1}^{N_U} p_{ik} \sum_{k=1}^{N_U} s_k}} \tag{1}$$

If a hierarchical Web site structure should be taken into account, then a modification of the cosine similarity, introduced in [3,4], that can take into account the Website structure can be used to yield the following input membership,

$$S_{si}^{web} = \max\left\{ \frac{\sum_{l=1}^{N_U}\sum_{k=1}^{N_U} p_{il} S_u(l,k) s_k}{\sum_{k=1}^{N_U} p_{ik} \sum_{k=1}^{N_U} s_k}, S_{si}^{cosine} \right\} \tag{2}$$

where $\boldsymbol{S_u}$ is a URL to URL similarity matrix that is computed based on the amount of overlap between the paths leading from the root of the website (main page) to any two URLs, and is given by

$$S_u(i,j) = \min\left(1, \frac{|p_i \cap p_j|}{\max(1, \max(|p_i|, |p_j|) - 1)}\right) \tag{3}$$

We refer to the special similarity in (2) as the *Web Session Similarity*.

3.1.2 Single-Step Decision-Tree Based Profile Prediction Recommender System

The nearest profile prediction model makes the critical assumption that sessions in different profiles are linearly separated. While this may be applicable for certain web mining methods, it may not be true for others. In order to be able to reliably map new unseen sessions to a set of mined profiles, without such assumptions about the profiles or how they separate the sessions, we can resort to classification methods that are not based on distance or similarity computations. In this paper, we explore both decision trees and neural networks for this task. Once

trained, using the decision tree or neural network model to classify a new session is fast, and constitutes the single step of the recommendation process, since the classified profile *is* the recommendation set.

The decision tree profile prediction model is very similar to the nearest profile prediction model. An input binary vector is presented as input to the decision tree [22] and a profile/class is predicted as the output. Each URL in the input vector is considered as an attribute. In learning, first the entire training data set is presented. An attribute value is tested at each decision node with two possible outcomes of the test, a branch and a sub-tree. The class node indicates the class to be predicted. An example is illustrated in figure 2.

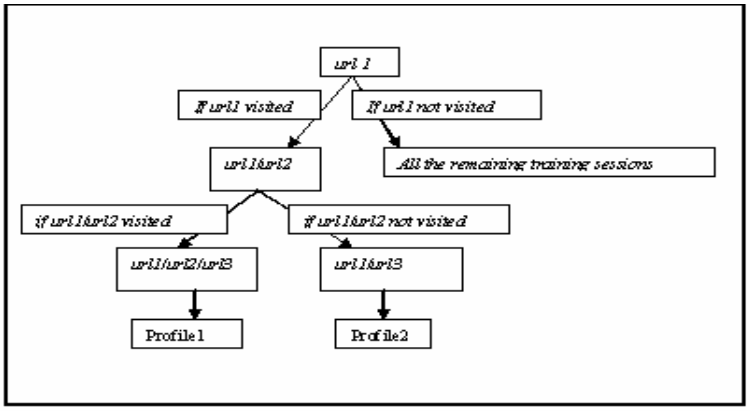


Fig. 2. Example of a Profile Prediction Model based on a decision tree that can be used within *CSA-1-step-Rec*

3.1.3 Single-Step Neural Network Based Profile Prediction Recommender System

In the neural network [21] based approach of profile prediction, a feed-forward multilayer perceptron is used and is trained with Back-Propagation. The inputs (session URLs) and output (class or profile) to the prediction model remain the same as the ones described above. The neural network replaces the classification model block in Figure 1. Hence the input layer of the network consists of as many input nodes as the number of valid URLs (i.e. N_U nodes), an output layer having one output node for each profile (i.e. N_p nodes), and a hidden layer with $(N_U + N_p)/2$ nodes. Figure 3 shows the architecture of the neural network used to predict the most relevant profile. The index of the output node with highest activation indicates the final class/profile.

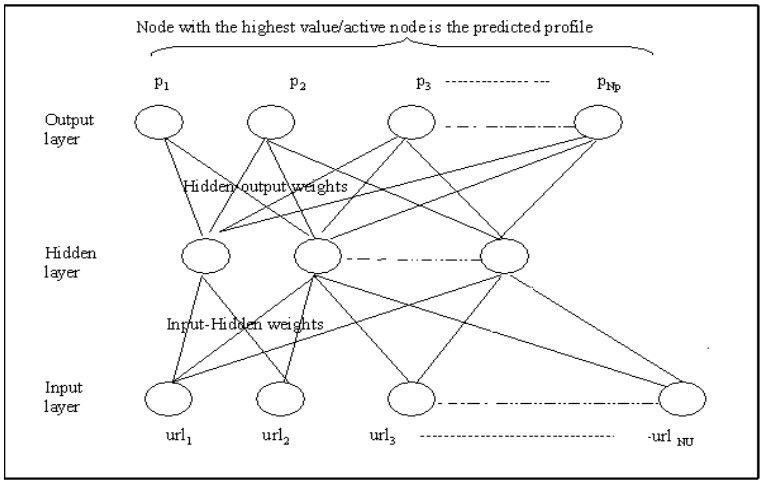


Fig. 3. Architecture of a Profile Prediction Model based on a Multi-Layer Perceptron that can be used within *CSA-1-step-Rec*

3.2 Context Ultra-Sensitive Approach Based on Two-Step Recommender System with A Committee Of Profile-Specific URL-Predictor Neural Networks (*CUSA-2-step-Rec*)

The single-step Profile prediction recommendation procedure is intuitively appealing and simple. In particular, its implementation and deployment in a live setting is very efficient. Essentially, it amounts to a look-up table. However, it has several flaws: ***(i)*** the degree of similarity between the current session and the nearest profile that is identified may not be taken into account, ***(ii)*** the above procedure does not take into account sessions that are similar to more than a single profile, ***(iii)*** it cannot handle sessions which are different from all known profiles, and ***(iv)*** the set of recommendations derive directly from the contents of a single (assigned) profile for all sessions assigned to this profile, without any further distinction between the specific access patterns. For this reason, we propose a two-step approach that in addition to exploiting the profile information, is able to recommend more highly personalized recommendations that depend not only on the assigned profile (*people-to-people* collaboration filtering), but also explicitly, on the input session itself (*item-to-item* collaboration filtering),.

3.2.1 Description of the Multi-Layer Perceptron URL-Predictor Neural Network

A Multilayer Perceptron neural network [21] can be used to predict the recommendation *URLs*. The architecture of this network, shown in Figure 4, is different from the network used in the *profile* prediction scenario of Figure 3. This is because the number of output nodes is now equal to the number of input nodes. The neural network is trained to complete the missing URLs of several complete ground-truth sessions, given as input several incomplete subsessions. This learning is analogous to completing some missing parts of a puzzle, as illustrated in Figure 12. Each training input consists of a user sub-session (*ss*) derived from a ground-truth complete session *S*, while training by example teaches the network output nodes to conform to the remainder of this session (*S*-*ss*). This means that there is one output node per URL. Hence, the architecture of the network can become extremely complex, as there would be N_U input and N_U output nodes. Training such a network may prove to be unrealistic on large websites that may consist of thousands of URLs. To overcome this problem, a separate network is learned for each profile *independently*, with an architecture of its own. The number of input and output nodes depends only on the number of significant URLs in that profile, and possibly those related to its URLs by URL-level or conceptual/semantic similarity. The number of hidden nodes is set to the average of number of input and output nodes. Figure 4 shows the architecture of each URL-predictor neural network. There will be a committee of N_p specialized networks of similar kind used in developing this URL recommendation prediction model, as illustrated in Figure 5. Each of these networks is completely specialized to forming the recommendations for a single profile, hence offering a local, more refined model, that enjoys the advantages of better accuracy, simplicity (fewer nodes and connections), and ease of training (as a result of simplicity).

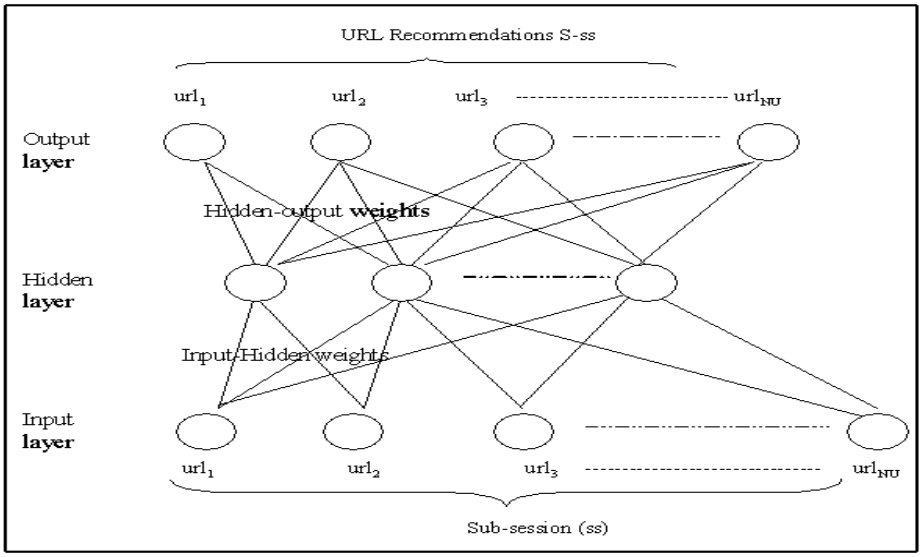


Fig. 4. Architecture of a Profile-Specific URL-Predictor Neural Network used in *CUSA-2-step-Rec*

3.2.2 Learning the Profile-Specific URL-Predictor Neural Network Models

The URL-Predictor network for each profile is learnt independently with a separate set of training data. Learning each network involves presenting a *sub-session* consisting of some of the URLs visited by the user belonging to that profile as input and adjusting the network weights by *back propagation* to recommend URLs that are not part of the sub-session given as input, but which are a part of the ground truth complete session, as output of the network. For each ground truth complete session, we find all the sub-sessions for window sizes 1-10, and use them to generate independent training and testing sets. Cosine similarity is used to map each sub-session to the closest profile, and the URL-Predictor network specialized for that profile is invoked to obtain the recommendations. A URL is considered to be recommended if its activation value exceeds a *'0.5'* at the corresponding output node of the invoked network.

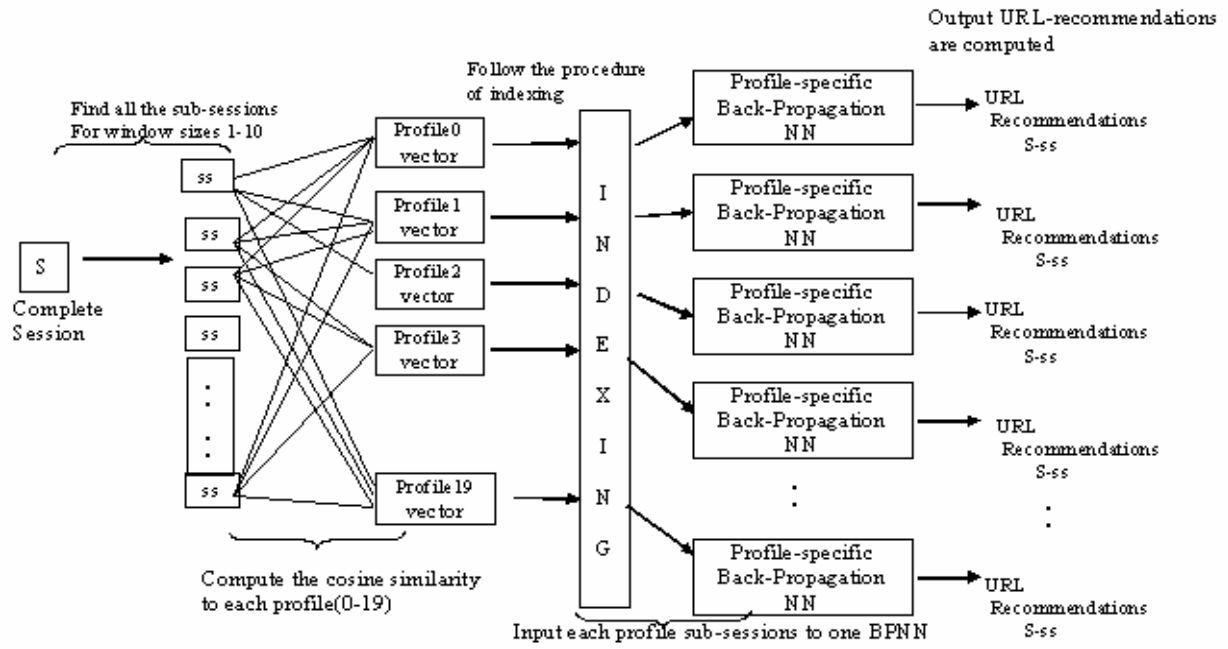


Fig. 5. Context Ultra-Sensitive Approach based on Two-Step Recommendation Process *(CUSA-2-step-Rec)* using a Committee of Profile-Specific URL-Predictor Neural Networks (Any URL-Predictor model can be substituted for the Multi-Layer Perceptron, e.g. a Hopfield network)

3.3 Recommendations Based On Autoassociative Memory Hopfield Networks

Hopfield networks are a special kind of recurrent neural networks that can be used as associative memory [21]. A Hopfield network can retrieve a complete pattern stored through the training process from an imperfect or noisy version of it. In some sense, a recommender system performs a similar operation, when it recommends certain URLs from an incomplete session. Given N_{url} fully connected (via symmetric weights w_{ij} between each two units i and j) neurons, each serving simultaneously as input and as output, and assuming that the activation values, x_i, are bipolar (+1/-1), the optimal weights to memorize N_p patterns, can be determined by *Hebbian* learning as follows

$$w_{ij} = \sum_{p=1}^{N_p} x_i^p x_j^p \qquad \text{for all } i \neq j \text{ (0, otherwise)} \tag{4}$$

During testing/recall, when a new noisy pattern x_{new} is presented as input, we set the activation at node i at iteration 0 to be $x_i^0 = x_{new\text{-}i}$, then the units are adjusted by iteratively computing, at each iteration t

$$x_i^{t+1} = \sum_{j=1}^{N_{url}} w_{ij} x^t_j \tag{5}$$

until the network converges to a stable state. However, the desired behavior of recall in a Hopfield network is expected to hold only if all the possible complete session prototypes can be stored in the Hopfield network's

connection weights, and if these complete sessions do not interact (or *cross-talk*) excessively. Severe deterioration starts occurring when the number of patterns exceeds a certain fraction of the number of nodes:

$$N_p > 0.138 N_{url}, \tag{6}$$

hence limiting a Hopfield recommender system to sites with a large number of URLs and yet very little variety in the user access patterns. This limitation is paradoxical in the context of large websites or transactional database systems. Our preliminary simulations with both a single global Hopfield network as well as several profile-specific Hopfield networks have resulted in low recall qualities since the network seemed to be able to memorize only very few stable states. However several profile-specific Hopfield networks perform better than one global network, but only for some of the profiles.

4 Experimental Results

4.1 Mining User profiles from Anonymous Web Usage Data

1703 web sessions accessing 343 URLs, extracted from log files of a university Web server, were used to generate training and testing sets. For each *complete* session considered as the *ground-truth*, all possible *sub-sessions* of different sizes are generated. The test dataset forms an independent 20% of the sub-sessions. Hierarchical Unsupervised Niche Clustering (H-UNC) [2] partitioned the web sessions into 20 clusters, each characterized by one of 20 profile vectors that were thoroughly checked and validated for consistency.

4.2 Comparative Simulation Results for *CUSA-2-step-Rec*, *CUSA-2-step-Rec*, and *K-NN* Collaborative Filtering

We used the following parameters in training the multilayer perceptron URL-Predictor neural networks: Maximum number of epochs = 2000, Learning Rate = 0.7 (for Input to Hidden layer) and 0.07 (for Hidden to Output layer), and a Momentum factor of 0.5. The Collaborative filtering approach is based on using K Nearest Neighbors (K-NN) followed by top-N recommendations for different values of K and N. First the closest K complete sessions from the entire history of accesses are found. Then the URLs present in these top K sessions are sorted in decreasing order of their frequency, and the top N URLs are treated as the recommendation set. We show only the best results obtained for K-NN at K=50 neighbors and N=10 URLs.

Figures 6 and 7, depicting the 20-profile averaged precision and coverage measures, show that the two-step profile-specific URL-predictor multilayer perceptron neural network recommender system *(CUSA-2-step-Rec)* wins in terms of *both* precision and coverage, particularly *above input sub-session size 2*. Figure 9 depicts the average *F1* measure, which is an equal aggregation of precision and coverage, for each input sub-session size. It may at first appear unusual that a recommendation strategy scores highly on both precision and coverage, and that an increase in precision did not seem to compromise coverage in any way. However, by looking at the details of the design of the profile-specific URL-predictor neural network, we explain this relentless increase in precision by the fact that the neural network output is trained to predict only the URLs that the user has *not* seen before, i.e. '*S-ss*', where *S* is the *complete* session, and *ss* is the sub-session (URLs *visited* by the user). Clearly, as the sub-session size increases, more URLs are presented to the output of the neural network, making the prediction task easier, since fewer URLs need to be predicted compared to smaller input sub-sessions. Similarly, coverage increases, since with more input URLs, the neural network is able to predict more of the missing URLs to complete the puzzle. However, this does not happen at the expense of precision. On the contrary, giving more hints about the user in the form of more of the visited URLs makes the prediction task easier, and hence, will only result in more accurate predictions.

We notice that the single-step recommender systems *(CSA-1-step-Rec)* do not have this nice feature, i.e., precision and coverage will generally have opposing trends. The performance of k-NN fares competitively with all the single-step recommender strategies, but only for longer session sizes. This is not surprising, considering that k-NN can yield very accurate predictions, because it too is based on local context-sensitive models. However, k-NN is notorious for its *excessive computational and memory costs, at recommendation time*, in contrast to all the other investigated techniques. While lazy in the learning phase, involving nothing more than storing the previously seen cases, k-NN takes its toll during the recommendation phase, when it needs to compare a new session with all past cases to produce recommendations.

Figures 10 and 11 depict the *F1* measures for each profile separately obtained with *CUSA-2-step-Rec* with specialized multilayer perceptron neural networks and k-NN, repectively. These figures show that the prediction

quality may vary widely between different profiles, since the sessions in some profiles are noisier, and hence are more difficult to predict. We also note that some profiles do not generate any testing sessions beyond a certain size because of their particular session length distribution. Table 1 summarizes the characteristics of the session lengths for each profile. The median length for most profiles is larger than 5 and for six of the profiles (0, 3, 4, 5, 11, and 15), it is greater than 9. For these profiles, half of the sessions have length greater than or equal to 9. Moreover, because we generate a large number of subsession combinations from each session for testing, we end up with a reasonably large number of test sessions (in the hundreds), especially between session size 2 and 8. We notice from Fig. 10 and 11, that at longer session lengths (above 5), the F1 measure with *CUSA-2-step-Rec* -NN far exceeds that of k-NN. This can be explained by the fact that while the performance of k-NN eventually saturates and even starts decreasing beyond a certain session length, that of the *CUSA-2-step-Rec* –NN approach can only improve, since each specialized network is essentially trained to complete the missing pieces (URLs) of a complete session, when given as input only some of the pieces. This is illustrated in Figure 12. Hence, it is only natural in this context that when more pieces are shown, a specialized neural network is better able to predict the missing pieces. The degradation of precision that results from higher coverage in k-NN approaches is avoided because the neural networks in *CUSA-2-step-Rec* are *trained* to be *precise*, while *excessive* coverage is controlled thanks to the *specialization* of each NN to only one of the profiles. Finally, we note that, if *all* input sub-session lengths are taken into account, then it is clear that a combination of *several different* recommender strategies, each applied only within its *optimal* range of sub-session length, will outperform each one of the recommender strategies acting on its own. In fact, in this case, even the very simple *CSA-1-step-Rec* strategy based on *nearest profile identification* outperforms all other strategies for very short input sessions (< 2 URLs). This is crucial to the retention and guidance of users who may be in their very initial browsing stages.

Finally, in Table 2, we show the performance (averaged over all session lengths) of the *CUSA-2-step-Rec* approach when specialized *Hopfield* networks are used for each profile instead of the *multilayer perceptron* neural networks. It is important to note that, while *testing* both types of neural networks was performed in a similar fashion, *training* them was a different matter. The Hopfield networks in our context are analogous to auto-associative memory banks. Hence, they were trained to memorize each complete session, and not to complete missing parts of a complete sessions from a large number of incomplete subsessions as in the multilayer perceptron neural networks.

We notice that while some profiles can be handled using the Hopfield networks, the performance for many profiles is poor, even sinking to complete failure for profiles 10, 17, 18, and 19. We attribute this failure to the excessive amount of cross-talk between the patterns to be memorized by the Hopfield networks for these profiles compared to the low number of nodes/URLs, especially in light of the constraint in (6). For example, as shown in Table 1, the Hopfield network for profile 18 had to memorize a large number of patterns: $N_p = 65$ training sessions in contrast with only $N_{url} = 5$ nodes. We have also trained a *single global* Hopfield network for all profiles to predict the URLs of incomplete sessions. Note that in this case, the constraint in (6) is severely violated with $N_p = 1703$ training patterns and $N_{url} = 343$ nodes. *Not surprisingly*, the average similarity between the memorized and retrieved sessions, obtained in this case, was *nil*.

Table 1: Number of URLs, sessions, minimum, maximum and median session lengths of each profile

profile	Number of Nodes (URLs)	Number of Sessions	Min Length	Max Length	Median Length
0	189	106	1	40	10
1	194	104	1	40	6
2	171	177	1	132	7
3	101	61	1	40	10
4	134	58	1	40	9
5	153	50	1	132	10
6	104	116	1	24	5
7	64	51	1	23	7
8	139	134	1	36	4
9	73	41	1	25	3
10	134	95	1	19	4
11	98	185	1	36	9
12	170	74	1	132	5
13	136	38	1	132	5
14	163	33	1	31	6
15	86	51	1	37	9
16	105	77	1	132	2
17	23	68	1	6	1
18	5	65	1	3	1
19	24	120	1	10	2

Table 2: Average cosine similarity between complete session and session retrieved from an incomplete input using several specialized Hopfield networks (one per profile). The similarity obtained when a *single global* Hopfield network is used for all profiles was *nil*.

Profile	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Similarity	.57	.4	.6	.18	.47	.13	.43	.62	.29	.31	0	.68	.26	.27	.28	.54	.30	0	0	0

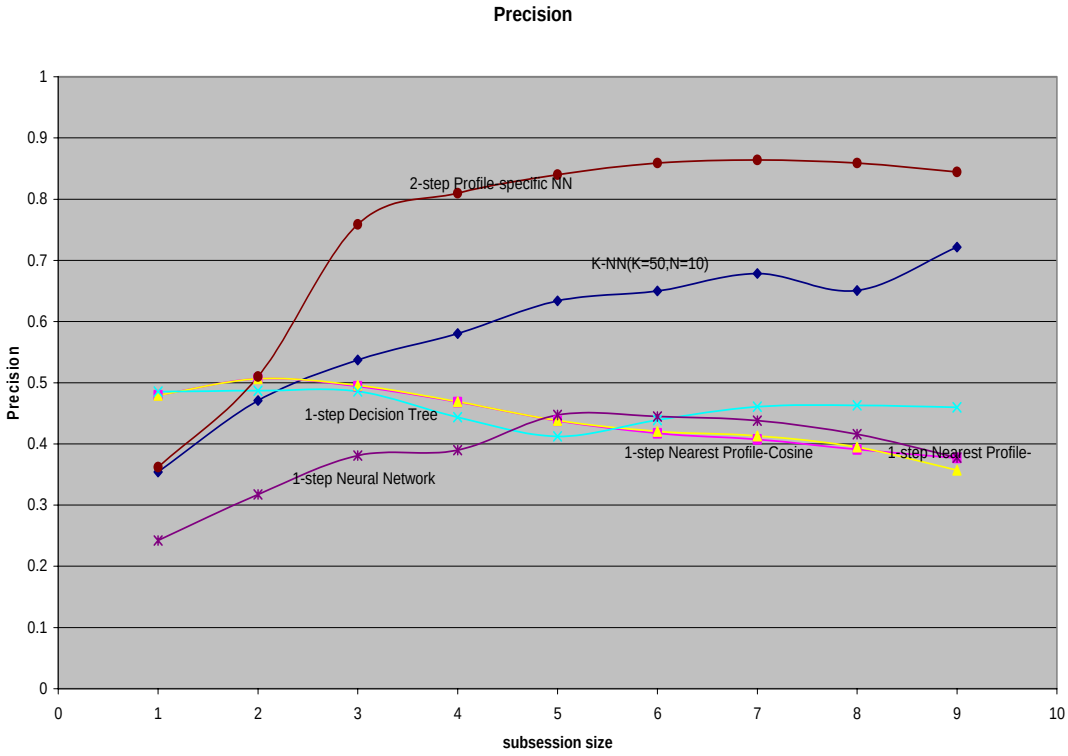


Fig. 6. Precision Values for all recommendation strategies (*CSA-1-step-Rec*, *CUSA-2-step-Rec*, and *K-NN*)

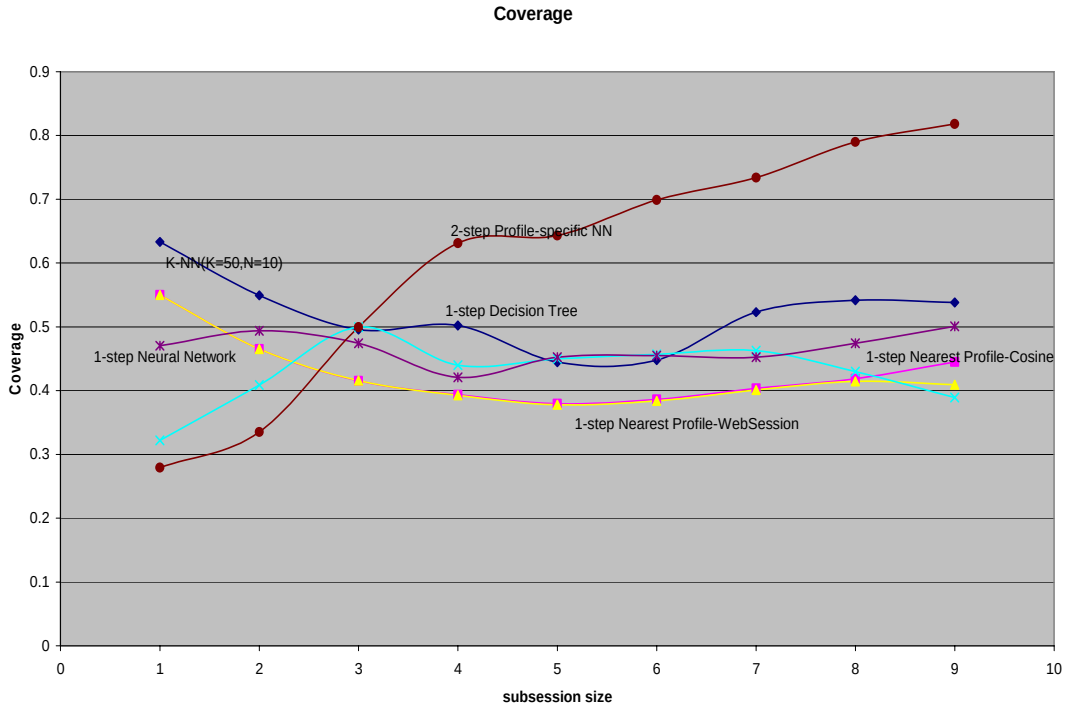


Fig. 7. Coverage Values for all recommendation strategies (*CSA-1-step-Rec*, *CUSA-2-step-Rec*, and *K-NN*)

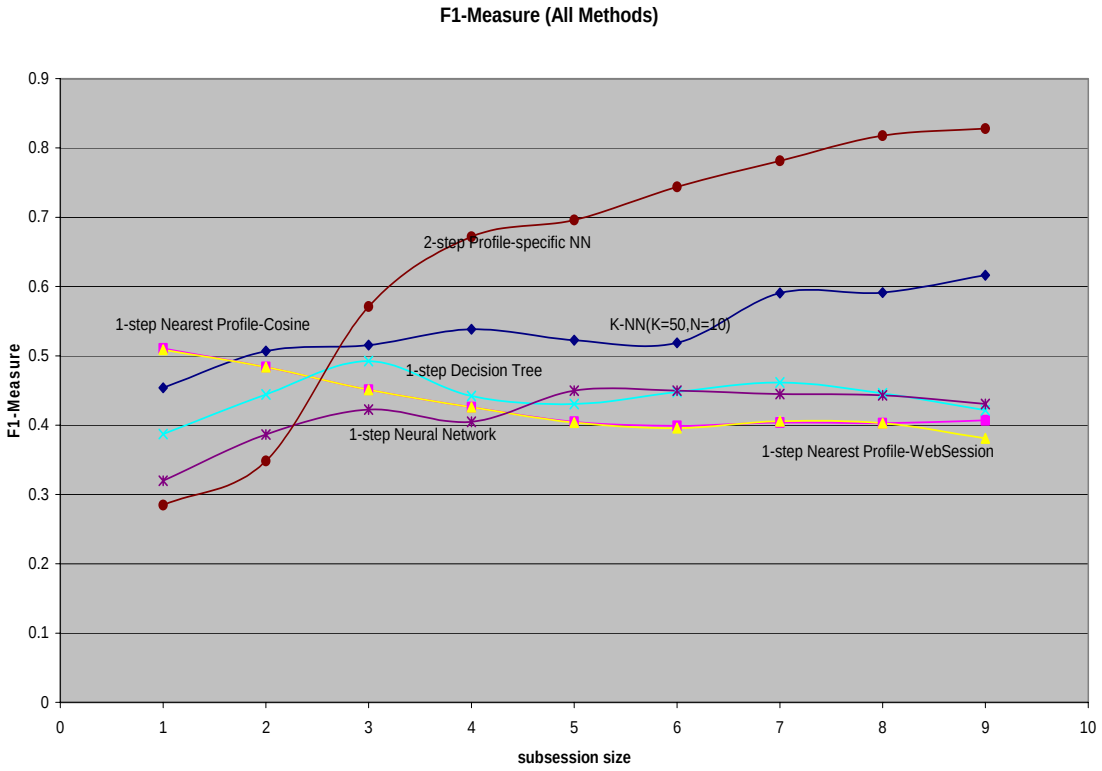


Figure 9: F1-Measure Values for all recommendation strategies (*CSA-1-step-Rec*, *CUSA-2-step-Rec*, and *K-NN*)

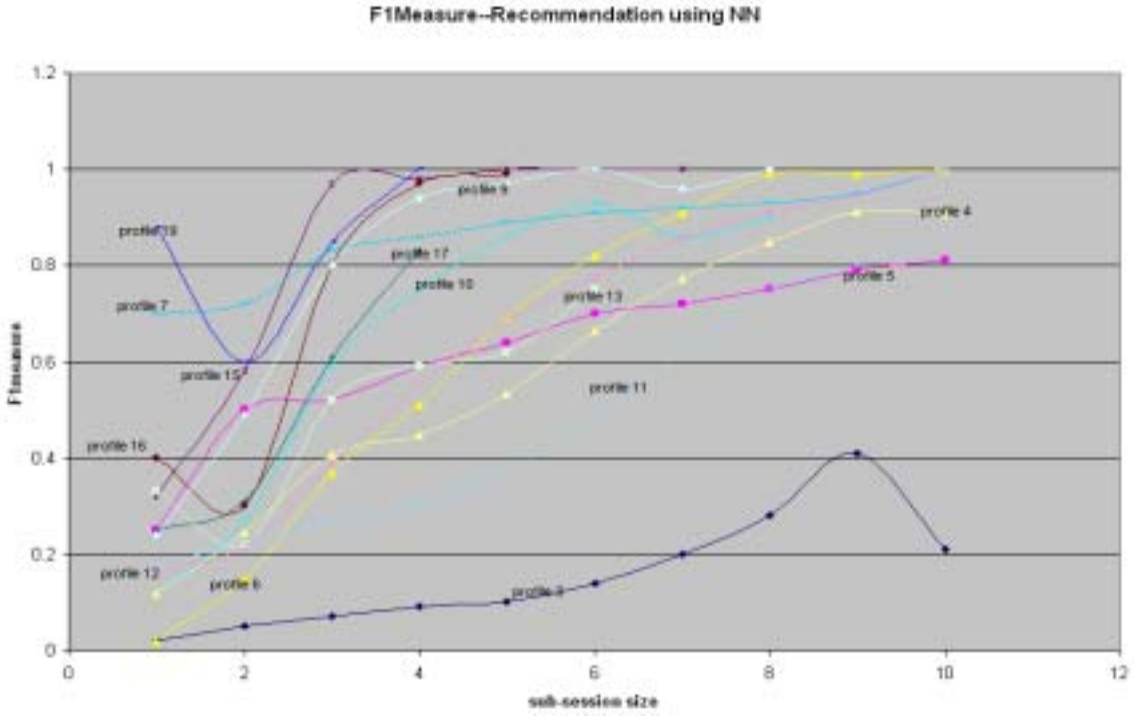


Figure 10: Individual averaged F1-Measure Values for each profile with the *CUSA-2-step-Rec* strategy

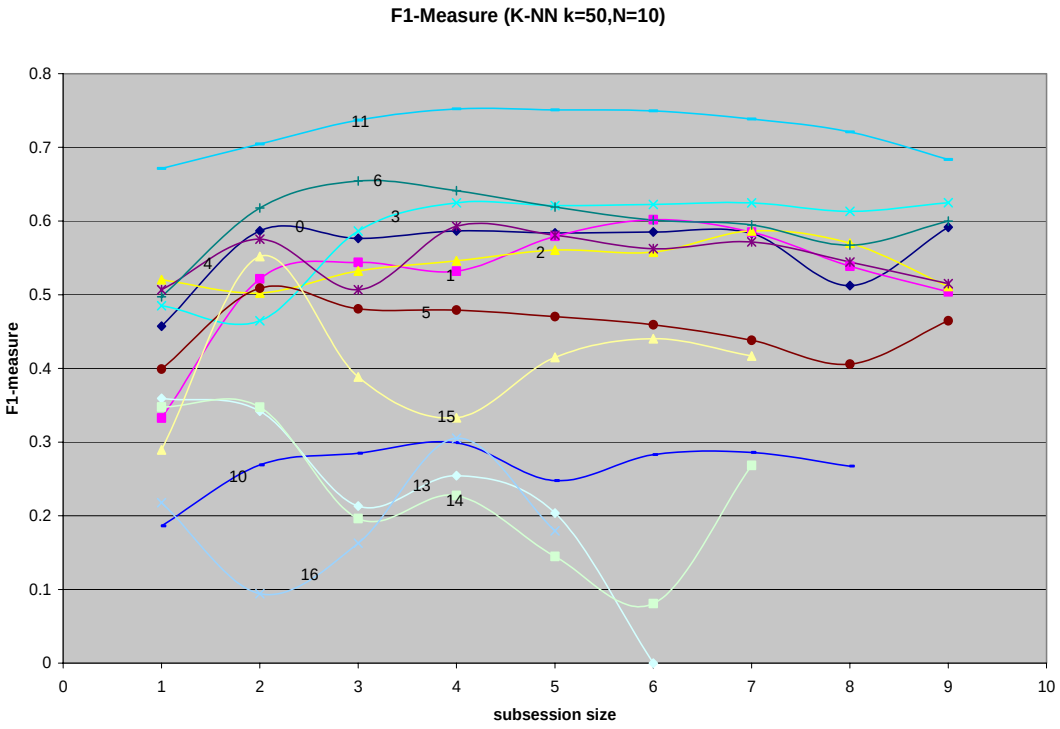
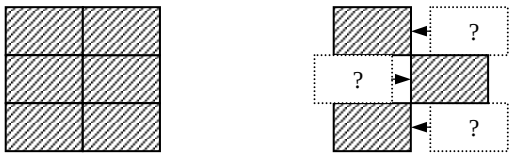


Figure 11: Individual averaged F1-Measure Values for each profile with the *K-NN* (K = 50, N = 10) strategy



(a) **(b)**

Figure 12: Completing the puzzle: (a) A complete session, (b) Input (incomplete) to the neural network (striped pieces) and output (marked with "?") that is predicted to complete the puzzle

5 Conclusions

We have investigated several single-step and two-step recommender systems. The single-step recommender systems *(CSA-1-step-Rec)* simply predict the URLs that are part of the nearest estimated profile as recommendations. The nearest profile prediction model simply based its recommendations on the closest profile based on a similarity measure, hence favoring linearly separable profile classes. In order to be able to reliably map new unseen sessions to a set of mined profiles, without such assumptions about the profiles or how they separate the sessions, we can resort to more powerful classification methods. In this paper, we explored both decision trees and neural networks for this task. Once trained, using the decision tree or neural network model to classify a new session constitutes the single step of the recommendation process, since the classified profile *is* the recommendation set. The two-step recommender system *(CUSA-2-step-Rec)* first maps a user session to one of the pre-discovered profiles, and then uses one of several profile-specific URL-predictor neural networks in the second step to provide the final recommendations. Based on this classification, a different recommendation model is designed for each profile separately. A *specialized* multilayer perceptron neural network was trained offline with back-propagation *for each profile* in order to provide a profile-specific recommendation strategy that predicts web pages of interest to the user depending on their profile. Each network was essentially trained to complete the *missing* pieces of several incomplete puzzles, with the pieces being the URLs, and each puzzle being a complete ground-truth session.

The *Hopfield* auto-associative memory network is an alternative to the multilayer perceptron that was also investigated. The Hopfield network is trained to memorize a complete session, then asked to retrieve this session when presented with only part of it. Our experiments confirmed that Hopfield networks can only form a reliable memory bank under severe constraints governing the relationship between the number of patterns to be memorized and the number of units in the network, and that unfortunately, these constraints are easily violated in typical real web usage environments. Nevertheless, *several profile-specialized* Hopfield networks in a *CUSA-2-step-Rec* framework performed significantly better than a *single* global network. The latter failed to form a reliable memory of the web usage patterns.

Unlike most previous work, the proposed two-step profile-specific URL-predictor neural network recommender system allows a more refined context sensitive recommendation process. The idea of using a separate network specialized to each profile seems to be novel, since it provides an even higher level of context-awareness in personalization than the level already offered through collaborative filtering based personalization. It is reasonable to expect that this modular design could be extended by replacing the URL-Predictor neural network modules by different learning paradigms that are faster to train, while not compromising the accuracy of predictions. The proposed model could also be made even faster to train and more accurate by encouraging the discovery of even more high-resolution profiles.

We finally classify our recommendation approaches with respect to the two-dimensional taxonomy presented in [16]. First, because the user is *anonymous* at all times, our approaches are all *ephemeral* with respect to the *persistence dimension*. Second, with respect to the *automation dimension*, our approaches are *fully automatic*. Furthermore, with regard to the four different families of recommendation techniques identified in [16] (*non-personalized, attribute based, item-to-item correlation, and people-to-people correlation*), the 1-step recommenders (*CSA-1-step-Rec*) can be considered as people-to people collaborative filtering. However, they use a cluster/profile summarization model, hence providing better scalability. On the other hand, the *CUSA-2-step-Rec* model uses people-to people collaborative filtering that is summarized through a cluster model, in the first stage to map a new user to a profile. Then it uses a specialized item-to-item recommendation model to produce the final recommendations. Therefore, the *CUSA-2-step-Rec* approach can be considered as a *hybrid* between *people-to-people* and *item-to-item* recommendations, and this fact, in addition to the quality of the preliminary Web usage mining results, may account for its good performance.

Acknowledgements

This work is supported by a National Science Foundation CAREER Award IIS-0133948 to Olfa Nasraoui.

References

[1] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically learning for user access pattern. Proc. 6th int. WWW conference, 1997.

[2] R. Cooley, B. Mobasher, and J. Srivastava, Web Mining: Information and Pattern discovery on the World Wide Web, Proc. IEEE Intl. Conf. Tools with AI, Newport Beach, CA, pp. 558-567, 1997.

[3] O. Nasraoui and R. Krishnapuram, and A. Joshi. Mining Web Access Logs Using a Relational Clustering Algorithm Based on a Robust Estimator, 8th International World Wide Web Conference, Toronto, pp. 40-41, 1999.

[4] O. Nasraoui, R. Krishnapuram, H. Frigui, and A. Joshi. Extracting Web User Profiles Using Relational Competitive Fuzzy Clustering, International Journal on Artificial Intelligence Tools, Vol. 9, No. 4, pp. 509-526, 2000.

[5] O. Nasraoui, and R. Krishnapuram, A Novel Approach to Unsupervised Robust Clustering using Genetic Niching, Proc. of the 9th IEEE International Conf. on Fuzzy Systems, San Antonio, TX, May 2000, pp. 170-175.

[6] O. Nasraoui and R. Krishnapuram. A New Evolutionary Approach to Web Usage and Context Sensitive Associations Mining, International Journal on Computational Intelligence and Applications - Special Issue on Internet Intelligent Systems, Vol. 2, No. 3, pp. 339-348, Sep. 2002.

[7] M. Pazzani and D. Billsus, Learning and revising User Profiles: The identification of Interesting Web Sites, Machine Learning, Arlington, 27, pp. 313-331, 1997.

[8] D.H. Kraft, J. Chen., M.J. Martin-Bautista, and M.A. Vila, Textual Information Retrieval with User Profiles Using Fuzzy Clustering and Inferencing, in *"Intelligent Exploration of the Web"*, Szczepaniak, P.S., Segovia, J., Kacprzyk, J., and Zadeh, L.A. (eds.), Physica-Verlag, Heidelberg, Germany, 2002.

[9] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, Effective personalizaton based on association rule discovery from Web usage data, ACM Workshop on Web information and data management, Atlanta, GA, Nov. 2001.

[10] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, 1975.

[11] L. Zadeh (1965). Fuzzy sets. Inf. Control 8, 338-353.

[12] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, 1995, ISBN 0-13-101171-5.

[13] R. Agrawal and R. Srikant (1994), Fast algorithms for mining association rules, Proceedings of the 20th VLDB Conference, Santiago, Chile, pp. 487-499.

[14] G. Linden, B. Smith, and J. York, *Amazon.com* Recommendations Iem-to-item collaborative filtering, IEEE Internet Computing, Vo. 7, No. 1, pp. 76-80, Jan. 2003

[15] J. Breese, H. Heckerman, and C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Proc. 14th Conf. Uncertainty in Artificial Intelligence, pp. 43-52, 1998.

[16] J.B. Schafer, J. Konstan, and J. Reidel, Recommender Systems in E-Commerce, Proc. ACM Conf. E-commerce, pp. 158-166, 1999.

[17] J. Srivastava, R. Cooley, M. Deshpande, and P-N Tan, Web usage mining: Discovery and applications of usage patterns from web data, SIGKDD Explorations, Vol. 1, No. 2, Jan 2000, pp. 1-12.

[18] O. Zaiane, M. Xin, and J. Han, Discovering web access patterns and trends by applying OLAP and data mining technology on web logs, in "Advances in Digital Libraries", 1998, Santa Barbara, CA, pp. 19-29.

[19] M. Spiliopoulou and L. C. Faulstich, WUM: A Web utilization Miner, in Proceedings of EDBT workshop WebDB98, Valencia, Spain, 1999.

[20] J. Borges and M. Levene, Data Mining of User Navigation Patterns, in *"Web Usage Analysis and User Profiling"*, Lecture Notes in Computer Science", H. A. Abbass, R. A. Sarker, and C.S. Newton Eds., Springer-Verlag , pp. 92-111,1999.

[21] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan, New York, 1994.

[22] J. R. Quinlan. Induction of Decision Trees. Machine Learning, Vol. 1, pp. 81--106, 1986.