Accurate Web Recommendations Based on Profile-Specific URL-Predictor Neural Networks

Olfa Nasraoui
Dept. of Electrical and Computer Engineering
206 Engineering Science Bldg.
The University of Memphis
Memphis, TN 38152-3180
(901) 678-3251
onasraou@memphis.edu

Mrudula Pavuluri
Dept. of Electrical and Computer Engineering
206 Engineering Science Bldg.
The University of Memphis
Memphis, TN 38152-3180
(901) 678-3251
pavuluri@memphis.edu

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. 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.

Categories & Subject Descriptors: H.2.8 [**Information Systems**]: Database Management: Database Applications – *data mining.*

General Terms: Algorithms, Performance, Design.

Keywords: Web mining, collaborative filtering, neural networks.

1. INTRODUCTION
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.

WWW 2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

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*)
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 (this is the frequency with which URL_k was accessed in the sessions belonging to the i^{th} cluster). The *recommendation* problem can be stated as follows. Given a current Web user session vector, $\boldsymbol{s}_j = [s_{j1}, s_{j2}, \ldots, s_{jN_U}]$, 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 *links* dynamically appended to the response to the most recent request. We denote the recommendations for current Web session, $\boldsymbol{s}_j$, by a vector $\boldsymbol{r}_j = [r_{j1}, r_{j2}, \ldots, r_{jN_U}]$, where $r_{jk} = 1$ if the k^{th} URL is recommended, and 0 otherwise. A multilayer perceptron neural network can be used to directly predict the URLs to be given as recommendations. Each training input consists of a user sub-session (*ss*) derived from a ground-truth complete session *S*, while the output nodes should conform to the remainder of this session (*S-ss*). This means that there is one output and one input 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. To overcome this problem, a separate network is learned for each profile independently. Thus, 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 conceptual similarity. There is a committee of specialized networks, one per profile, as illustrated in Figure 1. Each of these networks is specialized for one profile, hence offering a locally refined model. The network of each profile is learnt independently with a separate set of training data. Learning each network involves presenting the URLs visited by users belonging to that profile as input and learning the network weights that yield an activation value of 1 in the output units that correspond to the remaining URLs of the session, and 0 otherwise. At test-time, the output units generate URL *recommendations* after applying a threshold of *'0.5'* to the output activations.

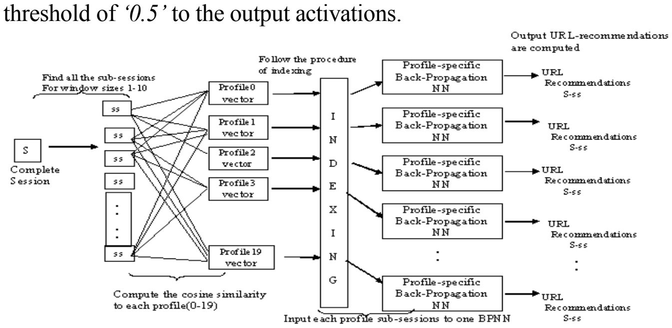

Figure 1: Recommendation Process based on a Committee of Profile-Specific URL-Predictor Neural Networks *(CUSA-2-step-Rec)*. Either Multi-layer Perceptrons or Auto-associative Memory Hopfield Networks can be used in the last stage)

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. 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

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

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

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

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 $N_p > 0.15N_{url}$, hence limiting 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.

4. SIMULATION RESULTS

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. We trained each neural network using *back propagation* with a maximum number of epochs = 2000, Learning Rate = 0.7 for the input to hidden layer, and 0.07 for the hidden to output layer, and Momentum = 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. Figure 3 shows the k-NN's 20-profile average of precision, coverage, and F1 measure (we show K=50, N=10, which gave the best results). A comparison with Fig. 2 shows that the profile-specific URL-predictor neural network recommender system wins in terms of both better precision and better coverage, particularly above session size 2. The k-NN also yields reasonably accurate predictions. However, k-NN is notorious for its *high computational and memory costs at recommendation time*.

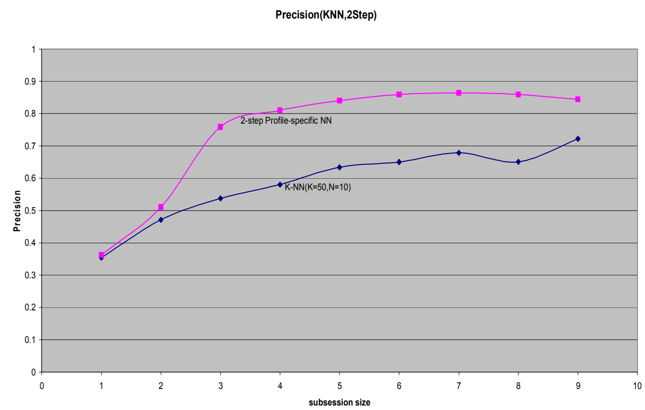

Figure 3: Average of F1-measure for the "*CUSA-2-step-Rec*" Profile-Specific URL-Predictor Recommender model (using Multilayer Perceptron Networks) compared to k-NN (k=50, top 10 recommendations)

5. CONCLUSIONS

Several *separate* and *smaller* URL-predictor neural networks, specialized to each profile, help provide a more refined context-aware personalization, while making training faster than global (all-profiles) neural network predictors because only the URLs relevant to one profile are used to define the network architecture. The *CUSA-2-step-Rec* is *hybrid*, mixing *people-to-people* (in step 1) and *item-to-item* (in step 2) recommendations, which may explain its superior performance.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

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

[2] Nasraoui O. and Krishnapuram R. "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, 339-348, Sep. 2002.

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