Proceedings of the 3rd Workshop on Intelligent Techniques for Web Personalization (ITWP'05), in conjunction with the 19th International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, Scotland, UK. August 1, 2005. (Bamshad Mobasher and Sarabjot Singh Anand, editors).

A Fuzzy Hybrid Collaborative Filtering Technique for Web Personalization

Bhushan Shankar Suryavanshi, Nematollaah Shiri, Sudhir P. Mudur
Dept. of Computer Science and Software Engineering
Concordia University, Montreal, Canada
{bs_surya,shiri,mudur}@cse.concordia.ca

Abstract

Collaborative filtering (CF) is a popular technique used in web personalization for building recommender systems, which predict the preferences of an active user, based on the preferences of past like-minded users. CF techniques are either memory-based or model-based. While the former is more accurate, its scalability compared to model-based is poor. An important contribution of this paper is a fuzzy hybrid CF technique that approaches the accuracy of memory-based and the scalability of model-based. This is achieved by innovatively utilizing properties of the underlying modeling technique used. The fuzzy nearest prototype of the active user is used to find a group of like-minded users within which a memory-based search is carried out. The group is small compared to the entire set, thus making the technique scalable. Results from comprehensive experiments using a large real life web log dataset and comparisons with implementations of memory-based and model-based techniques confirm our claim of accuracy and scalability.

1 Introduction

Web personalization is the process of customizing the content and structure of a web site to meet the specific needs of individual users, without requiring them to ask for it explicitly [Eirinaki and Vazirgiannis, 2003]. Its goal is to improve the user's experience of an e-service. For example, in e-commerce, personalization can be perceived as an online salesman. A salesman understands the needs of a customer through the initial interaction with the customer. Recognizing the user needs and meeting them successfully will surely lead to a long lasting, satisfying relationship and re-use of the services provided. Personalization can be compared to having your favorite bookseller pull out a copy of a book he just knows you would really like.

The web personalization process can be divided into four distinct phases [Mobasher *et al.*, 2000a; Nasraoui *et al.*, 2005], namely collection of web data, preprocessing of web data, analysis of web data, and finally recommendation which makes use of the results of the previous analysis step to provide recommendations to the user, such as adding hyperlinks to the last web page requested by the user, depending on the type of user. Recommendation engines could be based on content-based filtering, collaborative filtering, or rule-based filtering [Nasraoui *et al.*, 2005]. Content-based filtering exploits the product information, say, domain specific item attributes such as author and subject for book items, and artist and genre for music items. It does not require any previous implicit or explicit user rating or purchase data to make recommendations. In rule-based filtering, the user has to answer some questions, until s/he receives a customized result. It requires heavy planning, customization by expert, lacks intelligence, and tends to be static. Collaborative filtering (CF) is the most successful and widely used recommender system technology [Sarwar *et al.*, 2000b]. The goal of CF is to predict the preferences of a user, referred to as *active user*, based on the preference of a group of users. The key idea is that the active user will prefer those items that "like-minded" people prefer or even the ones that dissimilar people do not prefer. This approach relies on history, a dataset recording all previous users' interests, which could be inferred from their ratings of the items at a website (products or web pages). Rating can be *explicit,* such as previous purchases, customer satisfaction questionnaires, etc. or can be *implicit*, such as browsing activity on a website.

[Breese *et al.*, 1998] identified two major classes of collaborative filtering algorithms. *Memory-based* algorithms operate over the entire recorded user dataset of previous site usage to make predictions. These algorithms employ a notion of distance to find a set of users, known as neighbors, which tend to agree with active user. The preferences of neighbors are then combined to produce a prediction or top-N recommendation for the active user. *Model-based* algorithms on the other hand use the recorded dataset to estimate or learn a model, which is then used for predictions. Web usage mining techniques such as clustering, association rule mining, and sequence pattern discovery have been used for this purpose [Mobasher, 2004]. These techniques extract characteristics (patterns) from the user dataset through an offline process and employ these patterns to generate recommendations in the online process.

Two fundamental challenges in CF-based recommender systems are accuracy and scalability. Memory-based techniques are simple, provide high accuracy recommendations, and admit easy addition of new data. However, they are computationally expensive as the size of the input dataset increases. These techniques can be used to search tens of thousand of potential neighbors in real-time. But the demands of modern E-commerce systems are to search tens of millions of potential neighbors [Sarwar, 2000b]. On the other hand, model-based techniques reduce the online processing cost. This often comes at the cost of reduced accuracy of recommendation results. Moreover, the time complexity to compile the data into a model can very often be prohibitive [Pennock *et al.*, 2000; Pierrakos *et al.*, 2001].

In this paper, we propose a technique, which is a hybrid of memory-based and model-based CF approaches, inheriting the advantages of both. It is important to note that the term hybrid has been used in different senses in recommender systems earlier, as explained later in section 5. Our CF technique makes innovative use of the following two important properties of the model resulting from the underlying modeling technique, namely Relational Fuzzy Subtractive Clustering (RFSC) [Suryavanshi *et al.*, 2005], used in the offline learning phase. Firstly, the model has user sessions as cluster centers, giving us fuzzy cluster prototypes. Secondly, every session has varying degree of membership with each cluster. The fuzzy nearest prototype of the active user is used to find a group of like-minded users within which a memory-based search is carried out. This group is small in size compared to the original set, thus making the technique scalable. Furthermore, RFSC scales to large datasets and does not require any user specified control parameters.

The rest of this paper is organized as follows. Section 2 briefly describes data preparation phase and pattern discovery. In section 3, we present our fuzzy hybrid CF technique. Section 4 reports the results of our comprehensive experiments on real web log data. In section 5, we review related work. Section 6 includes conclusion and future work.

2 Data Preparation and Pattern Discovery

In the following, we briefly describe the tasks of preprocessing and cleaning of web log data. Interested readers are referred to [Cooley *et al.*, 1999; Mobasher, 2004] for details.

2.1 Session Identification

The access log from a web server is at a very fine granularity. Cleaning is required to remove log entries for image files and other such components within a web page. Also removed are log records for accesses by web crawlers and failed requests. A session is typically the URLs of pages visited by a user from the moment the user enters a web site to the moment the same user leaves it. Each distinct URL in the site is assigned a unique number j ranging from 1 to M, the total number of URLs. The collection of the user accesses in the k^{th} session is represented as a binary vector of size M, in which the j^{th} entry is 1 if the user accessed the j^{th} URL during this session, and is 0 otherwise.

2.2 Similarity Measure

Clustering techniques, which find groups in data, are used to extract *usage profiles* that capture different interests and trends among users accessing the site. Clustering can be done on object data or relational data. Numerical relational data [Hathaway *et al.*,1996] describes the set of objects to be clustered, less directly by giving a measurement of the dissimilarity (or similarity) between each pair of objects, and is represented by a matrix R, where R_{ij} is the dissimilarity between objects o_i and o_j. It also holds that $R_{ij}\geq 0$, $R_{ij}=R_{ji}$, and $R_{ii}=0$. Relational data clustering is useful when object dimensions are non-numeric.

For sessions, we use the similarity measure proposed in [Nasraoui *et al.*, 2000; 2002]. Session similarity in turn is defined based on URL similarity. The syntactic similarity between the i^{th} and j^{th} URLs is defined as follows:

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

where p_i denotes the path traversed from the root node (the main page) to the node which corresponds to the i^{th} URL. The length of path p_i is denoted as $|p_i|$.

In defining similarity of any two sessions s_k and s_l, two measures may be used. The first measure is *cosine*, which does not consider the site structure, and is defined as:

$$S_{1,kl} = \sum_{i=1}^{M} s_{ki}s_{li} \bigg/ \sqrt{\sum_{i=1}^{M} s_{ki} \sum_{i=1}^{M} s_{li}}$$

The second similarity measure, defined below, incorporates syntactic URL similarity.

$$S_{2,kl} = \sum_{i=1}^{M}\sum_{j=1}^{M} s_{ki}s_{lj}S_u(i,j) \bigg/ \sum_{i=1}^{M} s_{ki} \sum_{j=1}^{M} s_{lj}$$

The similarity between any pair of sessions *k* and *l* is a value in [0,1] defined as:

$$S_{kl} = \max(S_{1,kl}, S_{2,kl}) \qquad (1)$$

It follows that the dissimilarity between sessions *k* and *l* is:

$$D_{kl} = 1 - S_{kl} \qquad (2)$$

2.3 Pattern Discovery

The browsing behavior of users on the web is highly uncertain. A user might browse the same page for different purposes. Each time the user accesses the site, he/she may have different browsing goals. The same user in the same session may have different sub-goals and interests. This suggests that it is perhaps less meaningful to use crisp classes to capture these overlapping interests of users. To deal with this fuzziness and uncertainty, [Nasouri *et al.*,2000] proposed to extract profiles using an unsupervised relational clustering technique based on the competitive agglomeration algorithm. This idea is further extended in [Nasouri *et al.*,2002] by fuzzy clustering algorithms such as Relational Fuzzy C-Maximal Density Estimator (RFC-MDE) and Fuzzy C Medoids algorithm (FCMdd). All these techniques convert non-Euclidean relations into Euclidean by using the expensive β-spread transformations [Hathaway and Bezdek, 1994] which adds a positive number β to all off-diagonal elements of the relational matrix R. The value β should be chosen as small as possible to avoid loss of cluster information due to unnecessary spreads of data. The exact computation of β involves expensive eigenvalue calculations. The performance of the algorithms using this transformation depends on this value and it could be so large that the structure in original relational matrix R might not be mirrored by the transformed R [Corsini *et al.*, 2004]. Also, the success of these techniques depends on a number of "carefully" specified user input parameters such as number of clusters C, fuzzifier m, and initial partition U(0). It is not always possible to have *a priori* knowledge of these parameters for large datasets such as web logs. Due to such user control parameters, these clustering techniques may not manifest the true structure or groups in data. To overcome these difficulties, we proposed Relational Fuzzy Subtractive Clustering (RFSC) [Suryavanshi *et al.*, 2005] which is a highly scalable technique for extracting usage profiles. It does not require any user specified parameters, works well on large datasets, and also reduces the concern over the prohibitively long time taken for compiling the data into a model. Besides, RFSC is relatively more immune to noise. This is important when dealing with web usage data which is inherently noisy in nature. We also proposed a cluster validity index for RFSC to validate the clustering.

RFSC algorithm starts by considering each session as potential cluster center. The potential of any session s_i is calculates using the formula:

$$P_i = \sum_{j=1}^{N_U} e^{-\alpha R_{ij}^2}, \text{ where } \alpha = 4/\gamma^2$$

in which R_{ij} is the dissimilarity between sessions s_i and s_j, N_U is the total number of sessions to be clustered , γ is essentially the neighborhood, and is calculated from the relational matrix R as in [Suryavanshi *et al.*, 2005]. The session with the highest potential is selected as the first cluster center. Now, potential of each session is subtracted according to its dissimilarity with this cluster center. It follows that there is a larger subtraction in potential of sessions that are closer to the cluster center compared to those which are farther away. After this subtractive step, the session with the next highest potential is selected as the next cluster center. This process of subtraction and selection continues until a termination condition is met. To achieve the best clustering, we use the cluster validity index suggested.

After finding C cluster centers, the membership of every session s_j with each cluster c_i is calculated as follows.

$$u_{ij} = e^{-\alpha R_{c_i j}^2}, \; i = [1..C] \text{ and } j = [1..N_U],$$

where $R_{c_i j}$ is the dissimilarity of the i^th cluster center with the j^th session s_j. When s_j itself is the cluster center, we have $R_{c_i j} = 0$ and the membership $u_{ij} = 1$.

3 A Fuzzy Hybrid CF Technique

A CF algorithm recommends items or pages to the active user based on the preferences of N_U users in the user database. Let U be the set of URLs and M=|U|. Let, $s_1,\ldots,s_{Nu}$ denote the user sessions in the user database. We have $s_{ij} = 1$ if the i^th user accessed the j^th URL, or else $s_{ij} = 0$. Let s_a be the active user session. Let $NA \subset U$ be all the URLs not yet accessed by the active user, for which we would like to provide predictions. A collaborative filter is a function f that takes as input all past user sessions, and produces as its result recommendation values for pages not yet accessed by the active user [Pennock *et al.*, 2000]:

$$s_{aj} = f(s_1, s_2, \ldots, s_{Nu}), \qquad \forall\, j \in NA$$

Our goal in this work is to devise a CF technique whose accuracy is comparable to that of memory-based CF approach with scalability comparable to model-based approach. We use two important properties of model learnt from the RFSC algorithm, cluster centers and membership values. RFSC computes clusters whose centers are actual sessions. We call these cluster centers as *cluster prototypes*. If RFSC finds C clusters, then W= $\{Z_1,Z_2,\ldots,Z_C\}$ is the set of C prototypes representing these C clusters. The membership value u_{it} of each session s_t to cluster *i* is proportional to its distance or dissimilarity from the cluster center Z_i. The membership values of all the sessions that are clustered are stored in matrix u ($C \times N_U$). For an active session s_a, we first find the fuzzy nearest prototype [Keller *et al.*, 1985], i.e. the cluster p to which the membership u_{pa} is maximum. Now past like-minded user sessions of s_a will have their memberships close to u_{pa}, thus simplifying the computation of k-neighbors enormously. For these k-neighbors, we compute the URL popularity for only those URLs which are in NA. If the number of desired recommendations is N, then the top-N URLs, sorted in the order of their popularity, will be presented. The above steps are incorporated into the following algorithm.

Algorithm recommend:

```
Input: W -- set of prototypes;
       u ($C \times N_U$) -- membership matrix;
       $s_a$ -- active session;
Output: top-N recommended URLs.
Begin
/* STEP 1: find the fuzzy nearest prototype */
    Begin
        For i = 1 to C
            Calculate $D_{ia}$, the dissimilarity between $s_a$ and the
            i^th cluster prototype $Z_i$, using formula (2);
        EndFor
```

```
            find prototype $Z_p$ such that $D_{pa}$ is minimum;
            compute membership of $s_a$ to $p^{th}$ cluster using
            $u_{pa} = e^{-\alpha D_{pa}^2}$
        EndBegin
/* STEP 2: find k-nearest neighbors depending on membership */
        Begin
            Set k, a value in $\{1,\ldots,N_U\}$;
            Initialize neighbors = 0;
            For i = 1 to $N_U$
                If (neighbours $\leq$ k) Then
                    Include $s_i$ in the set of k-nearest neighbors;
                    Increment neighbors by 1;
                Else
                    Let $s_m$ be the farthest of the k-nearest neighbors
                    If ($| u_{pi} - u_{pa} | < | u_{pm} - u_{pa} |$)
                        Delete $s_m$ from the set of k-nearest neighbors;
                        Include $s_i$ in the set of k-nearest neighbors;
                    EndIf
                EndIf
            EndFor
        EndBegin
/* STEP 3: find the url popularity */
        Begin
            For All url j in NA
                Initialize $up_j = 0$;
                For All sessions $s_k$ in the set of k-nearest
                    neighbors
                    weight ($s_k$, $s_a$) = similarity between $s_a$ and $s_k$, using
                    formula (1);
                    $up_j = up_j + s_{kj}$ * weight ($s_k$, $s_a$);
                EndFor
            EndFor
        EndBegin
        Return top-N most popular URLs;
        EndAlgorithm
```

Fuzzy K- nearest prototype

In step one of the above algorithm, instead of finding the fuzzy nearest prototype, we could find fuzzy K-nearest prototype [Keller *et al.*, 1985]. Steps two and three can be carried out for each cluster in the same way, and at the end URL popularity scores from each cluster can be combined. Also if a URL is recommended from more than one cluster, then we consider maximum popularity score from all the contributing clusters. Results from our experiments show that this approach leads to increased accuracy of recommendation, but with slight increases in computation time.

Binning

In step two of finding k-nearest neighbors, we use equal-depth binning technique [Han and Kamber, 2000] to speed up the search of k-neighbors. Our algorithm uses the membership matrix u, which contains the memberships of each session to different clusters. Each row can be sorted in descending order of the membership values. Then this N_U interval is divided into b bins. Each bin contains N_U/b elements. We build a bin index for each row, which has b elements each of which contains value of membership at the beginning of each bin. Again as shown by our experimental results, use of this binning technique increases efficiency for finding the k-nearest neighbors with similar memberships.

4 Experimental Evaluation

In this section, we first present the evaluation metrics we have used for measuring recommendation quality, and follow it with a description of our experiments and the resulting values for these metrics.

4.1 Performance Evaluation Metrics

In a recommendation system, a possible measure for efficiency is the time taken by the system for producing an online recommendation, and for effectiveness it could be prediction quality [Kim *et al.*, 2004]. Two quantities called as recall and precision have been widely used as measures of effectiveness [Sarwar *et al.*, 2000b; Breese *et al.*, 1998; Pennock *et al.*, 2000]. Given any dataset, we first divide it into two parts: the training set and the test set. For every session in the test set, we hide some pages in this session, called as the Hidden set. Our algorithm works on the training set and generates a set of recommendations, for each session in the test set. Let top-N denote the set of TOPN number of pages recommended.

Recall is a global measure that corresponds to the proportion of relevant recommendations that have been retrieved by the system, i.e., the proportion of resources in the hidden set that are correctly recommended. The value of recall tends to increase as TOPN increases.

$$\text{Recall} = |\,\text{Hidden} \cap \text{top-N}\,| \,/\, |\,\text{Hidden}\,|$$

Precision measures the average quality of an individual recommendation. As TOPN increases, the precision of each recommendation decreases.

$$\text{Precision} = |\,\text{Hidden} \cap \text{top-N}\,| \,/\, |\text{top-N}\,|$$

A measure which combines recall and precision with equal weights has also been suggested and is defined as follows:

$$\text{F1} = (2 \times \text{recall} \times \text{precision}) \,/\, (\text{recall} + \text{precision}).$$

4.2 Experiments and Results

For our experiment, we have used the user access logs from the web server of Computer Science and Software Engineering Department (CSE) at Concordia University during the period of June 15, 2004 to July 5, 2004. We applied the preprocessing step [Cooley *et al.*, 1999] and created sessions by considering 45 minutes as the maximum elapsed time between two consecutive accesses from a single IP address. Root “/” was filtered out from all sessions as it appeared in more than 80% of the sessions. We also removed short sessions of length 1 or 2. After this preprocessing phase, we obtained 12,227 user sessions and 10,153 distinct URLs. The average length of the sessions was 8.54 pages and sparsity level was 0.999312, defined as 1- (nonzero entries / total entries). We then divided (randomly) this dataset of user sessions into (1) the training set, with 10,000 sessions (81.7%) and (2) the test set, with 2227 sessions (18.3%).

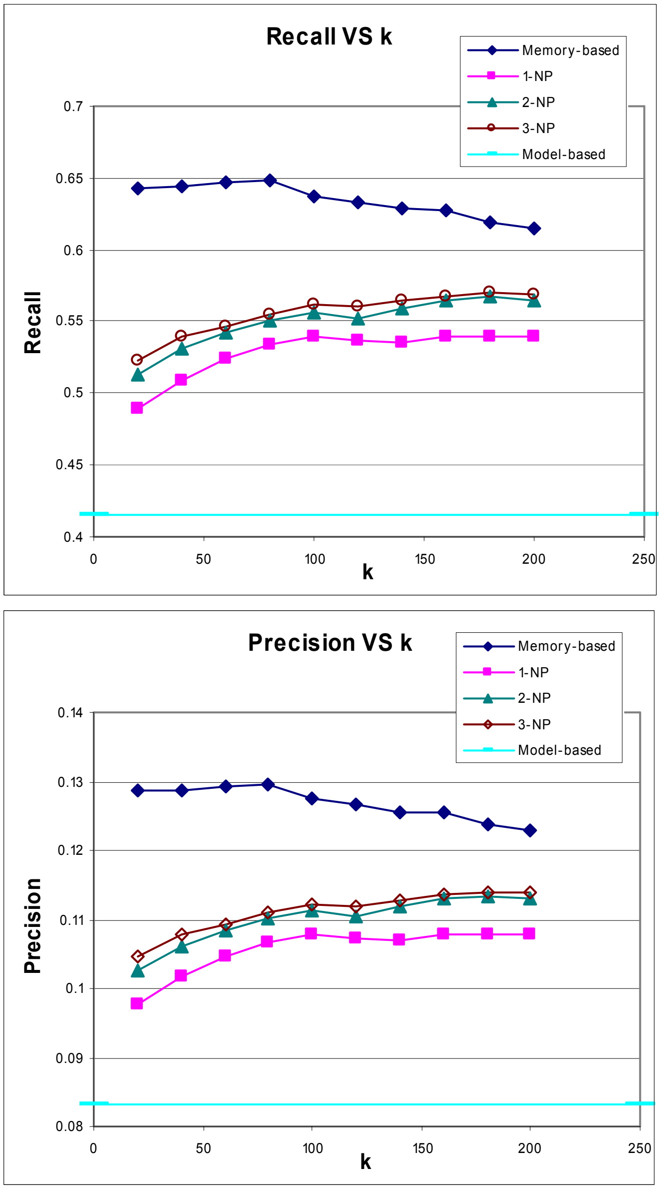

Figure 1. Impact of neighborhood size k on recommendation quality.

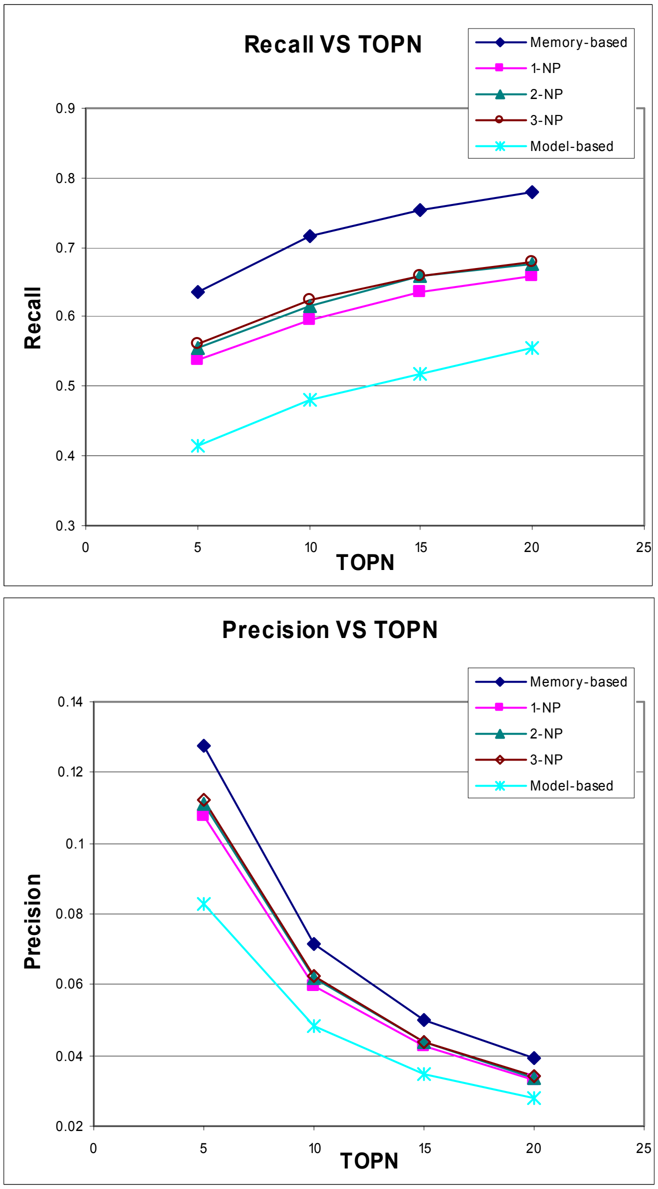

Figure 2. Impact of TOPN on recommendation quality.

Using the RFSC algorithm described earlier and the cluster validity index introduced in [Suryavanshi *et al.*, 2005] we obtained the clusters. In all, 34 clusters were found, i.e., C=34. In our experiment we have compared the predictive ability of our fuzzy hybrid collaborative filtering approach with both memory-based (k-nearest neighbor) and model-based approaches. The clustering technique used for model-based approach is RFSC so as to enable fair comparison. We adopted the all-but-1 protocol used in [Breese *et al.*, 1998] by which one page is randomly withheld (hidden) from every session in the test set. We then applied each of the three algorithms, memory-based, model-based, and our fuzzy hybrid CF technique to get a set of recommendations for every session in the test set. Lastly, we checked to see if the hidden page was present in the top-N pages recommended by the respective algorithms. We carried out different experiments by calculating recall, precision, and F1 measures for different values of neighborhood k, number of fuzzy nearest prototype K, and TOPN.

Figure 1 shows the comparison of recommendation quality of different approaches of memory-based, model-based, and the hybrid (Fuzzy 1-Nearest Prototype (1-NP), 2-Nearest Prototype (2-NP), 3-Nearest Prototype (3-NP)).
We only show the recall and precision measures and F1 can be inferred from these two. The recall and precision measures for model-based are shown as lines parallel to the x–axis, since there is no notion of k in the model-based approach. For this experiment, we fix the value TOPN to 5. It

can be seen that the size of the neighborhood has a significant impact on the recommendation quality. The quality of memory-based increases up to a certain point as k is increased, after which the quality starts decreasing. For 1-NP, 2-NP, and 3-NP, the quality increases with k and remains almost constant after some point. Also quality increases for 2-NP and 3-NP in that order. The effectiveness of our hybrid approach is much better than model-based and comparable with memory-based for higher values of k.

Figure 2 shows the comparison of different approaches by keeping k constant at 100 and varying TOPN. In general, recall increases as TOPN increases while precision decreases with increase in TOPN. For higher values of TOPN, all the three approaches tend to yield the same precision.

In Figure 3, we show the comparison of efficiency in terms of online recommendation time. As can be seen, the time taken for memory-based approach is much higher than our hybrid approach. On average, time taken for memory-based approach is 30 to 40 times more than the time taken by the hybrid approach. Also for 1-NP, 2-NP, and 3-NP the time increases in that order, but at the same time, this small increase in time results in increased recommendation quality. The average time taken by model-based approach was noted to be around 0.168388 millisecond/user. On an average, the time taken by the hybrid approach is just about 3 to 5 times more than the model-based approach.

Figure 4 shows comparison of improvement in recommendation time by applying equal-depth binning to speed up the search for the k neighbors in our hybrid approach. The binning method improved the efficiency by a factor of 6, on the average.

5 Related Work

Collaborative filtering has been studied extensively in e-commerce. The GroupLens system [Konstan *et al.*, 1997] is amongst the earliest CF systems. It is a recommender system based on the Usenet news groups wherein recommendations are made according to the correlations among the news ratings provided by the users. For surveys of various collaborative filtering algorithms, interested readers are referred to [Breese *et al.*, 1998; Sarwar *et al.*, 2000b]. To improve performance of recommender systems, [Sarwar, 2000a] considers dimensionality reduction technique. Clustering is successfully used to summarize and analyze data in many different domains. As a natural extension, clustering has been used in various recommender systems [Ungar and Foster, 1998; Breese *et al.*,1998; O'Connor and Herlocker, 1999] by grouping customers with similar profiles into the same cluster and then recommending products based on the popularity of each product within that cluster.

Most work in Web usage mining has focused more on producing analytical knowledge rather than its use for personalization. [Perkowitz and Etzioni, 1997] introduced the notion of *adaptive Web* sites, defined as sites that semi automatically improve their organization and presentation by learning from visitors' access patters.

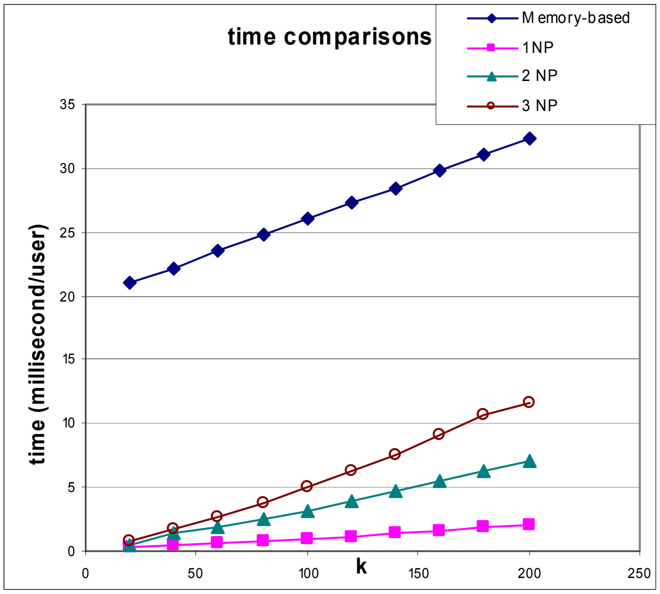

Figure 3. Comparison of efficiency in terms of online recommendation time.

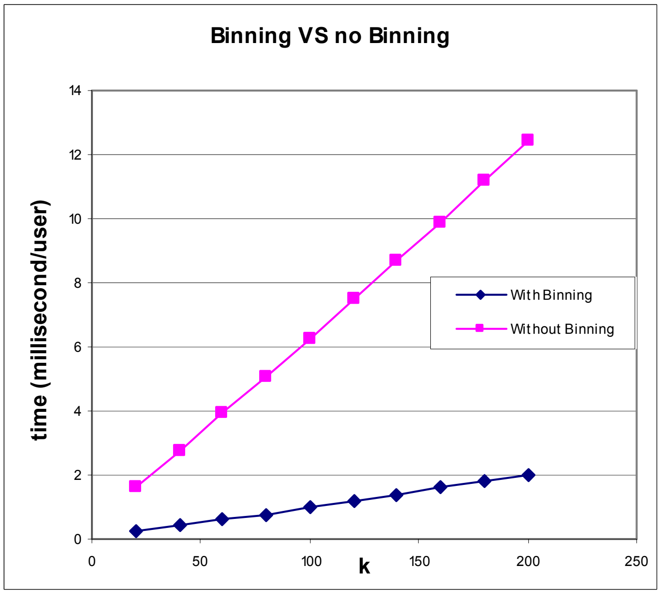

Figure 4. Comparison of efficiency (Binning)

[Mobasher *et al.* 1999; 2000a] proposed a comprehensive framework for mining Web logs to discover knowledge for providing recommendations to current users based on their browsing similarities to previous users. Web usage mining techniques such as association rules, sequential pattern discovery, clustering, and classification are applied in order to discover interesting usage patterns. The results are then used for the creation of aggregated usage profiles, in order to generate decision rules. The recommendation engine matches each user's activity against these profiles and provides him with a list of recommended hypertext links. This framework was further extended in [Mobasher *et al.*, 2000b] to incorporate content profiles into the recommendation process as a mechanism to enhance the effectiveness of personalization.

The term ‘hybrid’ has been used in different senses by different authors. [Shahabi *et al.*, 2001] proposed a recommender system which employs a hybrid approach that combines collaborative filtering and content-based querying to achieve better accuracy. A hybrid method based on Personality Diagnosis (PD) is proposed in [Pennock *et al.*,2000]. In this method, the personality type of the active user is determined and used to compute the probability of the active user requiring new items. PD-based collaborative filtering requires using all the available data throughout the process, though new data can be added incrementally to adjust the model parameters. [Nakagawa and Mobasher, 2003] also proposed a hybrid model utilizing the site structure and the degree of local hyperlink connectivity. [Kim *et al.*,2004] proposed a hybrid model for improving the performance by applying four prediction models – the Markov model, sequential association rule, association rule, and a default model in tandem in their precision order. Our sense of hybrid, as described, is to use both memory-based and model-based approaches for provision of more accurate predictions at reasonable increase in cost.

6 Conclusion and Future Work

Two fundamental challenges to CF-based recommender system are accuracy and scalability. While memory-based methods have a high accuracy, they become prohibitively expensive to compute as the size of the input dataset increases. The model-based approach has a complementary advantage: while it reduces the online processing cost, it often comes at the cost of reduced accuracy of recommendations. In this work, we have proposed a new fuzzy hybrid CF technique, whose accuracy is comparable to memory-based CF and whose performance is comparable to model-based approach. This was achieved due to two important properties of the RFSC technique used to generate the model in the offline phase. Firstly, the model has user sessions as cluster centers, giving us fuzzy prototypes. Secondly, every session in this model has varying degree of membership with each cluster. The fuzzy nearest prototype of the active user is used to find a group of like-minded users within which a memory-based search is conducted. We also extend our technique to fuzzy K-nearest prototypes (K-NP). We have implemented this technique and carried out comprehensive experiments on real life web log records, comparing the performance with results from memory-based and model-based approaches. Results from our experiments confirm that the accuracy of our technique is comparable to that of memory-based and it has the scalability of model-based approaches. Our use of Relational Fuzzy Subtractive Clustering has the added advantage that it works well on large datasets and also reduces the concern over the prohibitively large time taken for compiling the data into a model.

An interesting future extension would be to compare the recommendation quality of our approach with other model-based approaches which employ web usage mining techniques, such as association rule mining or sequence mining. Also the evaluation metrics such as recall, precision, and F1 are pessimistic or offline metrics; they only consider a recommendation to be correct if it exactly matches the hidden item [Veloso *et al.*, 2004]. The fact that a recommendation is not exact does not mean that it is inadequate. Though offline analysis is useful, we plan to evaluate user satisfaction with this recommender model in an online context by taking user’s feedback. We would like to study the impact of various preprocessing techniques which have been proposed [Mobasher *et al.*, 2001] to improve the recommendation effectiveness.

Acknowledgements

This work was partially supported by Discovery Grants from NSERC, and by ENCS, Concordia University Faculty Research Support Grants.

References

[Breese *et al.*, 1998] Breese, J., Heckerman, D., Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of UAI-98*, pp. 43-52, 1998.

[Cooley *et al.*,1999] Cooley, R., Mobasher, B. and Srivastava, J. Data Preparation for Mining World Wide Web Browsing Patterns. *J. of Knowledge and Information Systems*, 1, pp. 1-27,1999.

[Corsini *et al.*, 2004] Corsini, P., Lazzerini, B., Marcelloni, F. A new fuzzy relational clustering algorithm based on fuzzy C-means algorithm. *Soft Computing*, 2004, Springer-Verlag.

[Eirinaki and Vazirgiannis, 2003] Eirinaki M., Vazirgiannis M., Web mining for Web personalization. *ACM Transactions on Internet Technology* 3(1): 1-27, 2003.

[Han and Kamber, 2000] Han, J., Kamber, M., Data Mining: Concepts and Techniques, Morgan Kaufmann.

[Hathaway *et al.*, 1996] Hathaway, R.J., Bezdek, J.C., Davenport, J.W. On relational data version of c-means algorithm, *Pattern Recognition Letters*, 17, pp. 607-612, 1996.

[Hathaway and Bezdek, 1994] Hathaway, R.J., Bezdek, J.C. NERF c-means: Non-Euclidean relational fuzzy clustering, *Pattern Recognition*, 27, pp. 429-437, 1994.

[Keller *et al.*, 1985] Keller, J., Gray, M. and Givens, J., A fuzzy k-nearest neighbor algorithm, *IEEE Transaction on Systems, Man and Cybernetics*, 15(4): 580, 1985.

[Kim *et al.*, 2004] Kim, D., Il Im, Atluri, V., Bieber, M., Adam, N., Yesha, Y. A clickstream-based collaborative filtering personalization model: towards a better performance. In *Proc. of WIDM 2004*, pp. 88-95, Washington, DC, USA, 2004.

[Konstan *et al.*, 1997] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77-87, 1997.

[Mobasher, 2004] Mobasher, B. Web Usage Mining and Personalization, P*ractical Handbook of Internet Computing*, Munindar P. Singh (ed.), CRC Press, 2004

[Mobasher *et al.*, 1999] Mobasher, B., Cooley, R., Srivastava, J. Creating adaptive web sites through usage-based clustering of URLs. In *Proc. of KDEX'99*, Nov. 1999.

[Mobasher *et al.*, 2000a] Mobasher, B., Cooley, R., Srivastava, J. Automatic personalization based on web usage mining. *Commun. ACM*, 43, pp. 142–151, August 2000.

[Mobasher *et al.*, 2000b] Mobasher, B., Dai, H., Luo, T., Sung, Y., Zhu, J. Integrating web usage and content mining for more effective personalization. In *Proc. of the International Conference on Ecommerce and Web Technologies,* Greenwich, UK, Sept 2000.

[Mobasher *et al.*, 2001] Mobasher, B., Dai, H., Luo, T., M. Nakagawa. Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. In *Proc. of ITWP'01*, Seattle, August 2001.

[Nakagawa and Mobasher, 2003] Nakagawa, M., Mobasher, B. A Hybrid Web Personalization Model Based on Site Connectivity. In *Proc. of WEBKDD 2003*, pp. 59-70, Washington USA, August 28, 2003.

[Nasraoui *et al.*, 2005] Nasraoui, O. World Wide Web Personalization. *Encyclopedia of Data Mining and Data Warehousing*, J. Wang, Ed, 2005, Idea Group.

[Nasraoui *et al.*, 2000] Nasraoui O., Frigui H., Krishnapuram R., and Joshi A. Extracting Web User Profiles Using Relational Competitive Fuzzy Clustering. *International Journal on Artificial Intelligence Tools*, 9(4): 509-526, 2000.

[Nasraoui *et al.*, 2002] Nasraoui O., Krishnapuram R., Joshi A., and Kamdar T. Automatic Web User Profiling and Personalization using Robust Fuzzy Relational Clustering, in *E-Commerce and Intelligent Methods* Ed., 2002, Springer-Verlag.

[O'Connor and Herlocker, 1999] O'Connor, M. & Herlocker, J. Clustering Items for Collaborative Filtering. In Proc. of *the ACM SIGIR Workshop on Recommender Systems*, CA, 1999.

[Pennock *et al.*, 2000] Pennock, D. M., Horvitz, E., Lawrence, S., and Giles, C. L. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proc. of UAI-2000*, pp. 473-480, Stanford, CA, 2000.

[Perkowitz and Etzioni, 1997] Perkowitz, M., Etzioni, O. Adaptive web sites: An AI challenge. In *Proc. of the Fifteenth International Joint Conference on Artificial Intelligence,* Japan, 1997.

[Pierrakos *et al.*, 2001] Pierrakos, D., Paliouras, G., Papatheodorou, C., Spyropoulos, C., KOINOTITES: A Web Usage Mining Tool for Personalization, In *Proc. of the Panhellenic Conference on Human Computer Interaction*, Patras, December 2001.

[Sarwar *et al.,* 2000a] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. Application of dimensionality reduction in recommender system—A case study. In *Proc. of WebKDD 2000 Web Mining for e-Commerce Workshop*, Boston, USA, 2000.

[Sarwar *et al.*, 2000b] Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J. Analysis of recommender algorithms for e-commerce. In *Proc. of the 2nd ACM E-commerce Conference*, Minnesota, USA, 2000.

[Shahabi *et al.*,2001] Shahabi, C., Kashani, F., Chen, Y., McLeod, D. Yoda: An Accurate and Scalable Web-Based Recommendation System. In *Proc. of CoopIS 2001*, pp. 418-432, Italy, 2001.

[Suryavanshi *et al.*, 2005] Suryavanshi, B.S., Shiri, N., Mudur, S.P. An Efficient Technique for Mining Usage Profiles using Relational Fuzzy Subtractive Clustering. In *Proc. of WIRI' 05,* Tokyo, Japan, April 8-9, 2005.

[Ungar and Foster, 1998] Ungar, L. H. and Foster, D. P. Clustering methods for collaborative filtering. In *Proc. of the 1998 Workshop on Recommender Systems*, 1998, AAAI Press.

[Veloso *et al.* 2004] Miguel Veloso, Jorge, A., Azevedo, P. Model-Based Collaborative Filtering for Team Building Support. In *Proc. of ICEIS*, pp. 241-248, Portugal, April 2004.