A New Hybrid Recommender System Using Dynamic Fuzzy Clustering

Baghbani Mojtaba
Dept. of Computer Engineering
University of Mashhad
Mashhad, Iran
mo_ba97@stu-mail.um.ac.ir

Monsefi Reza
Dept. of Computer Engineering
University of Mashhad
Mashhad, Iran
rmonsefi@ferdowsi.um.ac.ir

Abstract: *In this paper[1], a new hybrid system is proposed for combining collaborative and content-based approaches that resolves some limitations of them. Specially, by the proposed system, the novelty and diversity of recommendations improve remarkably. Furthermore, the precision and recall of the proposed system is slightly less than those of the best existing hybrid system (collaborative via content) so that employing this system is justifiable. By this approach, the items that have not been yet rated by any user can be recommended. Collaborative and content-based systems utilized by this work, use a hybrid method based on fuzzy clustering model (fuzzy subtractive clustering) that combines model and memory-based approaches so that its precision is comparable with the precision of the memory-based approach and its scalability is comparable with the scalability of the model-based approach. Furthermore, in this work, a dynamic fuzzy clustering algorithm was proposed in which a measure is presented to determine the stage at which a complete reclustering is required. By applying this algorithm, the system is able to adapt to the dynamic and changing environment in a much less expensive manner in terms of computation times and resources.*

Keywords: Recommender system, Content-based recommender, Collaborative recommender, Hybrid recommender, Relational fuzzy subtractive clustering, Dynamic clustering.

1 Introduction

Recommender systems have recently gained much attention as a new business intelligence tool for e-commerce business [14]. Applying a recommender system for an online retailer store helps to enhance the quality of service for customers and increase the sale of products and services. The recent commercial success of recommender systems has been demonstrated in many online stores including Amazon.com, CDNow.com, Barnes&Noble.com, and MovieFinder.com [5].

Recommendation engines could be based on *content-based filtering* or *collaborative filtering* [8]. 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. Collaborative filtering (CF) is the most successful and widely used recommender system technology [13]. 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 an online store.

[3] identified two major classes of collaborative filtering algorithms. *Memory-based* algorithms operate over the entire recorded user dataset 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

[1] This work was supported in part by Iran Telecommunication Research Center (ITRC).

other hand use the recorded dataset to estimate or learn a model, which is then used for predictions.

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 [13]. 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 [9, 10].

There is an emerging understanding that good recommendation accuracy alone does not give users of recommender systems an effective and satisfying experience [6]. Recommender systems must provide not just accuracy, but also usefulness. For instance, a recommender might achieve high accuracy by only computing predictions for easy-to predict items – but those are the very items for which users are least likely to need predictions [6].

In this paper, we consider measures of recommender system usefulness that move beyond accuracy to include suitability of the recommendations to users. Suitability includes *coverage*, which measures the percentage of a dataset that the recommender system is able to provide predictions for; and novelty, which measure whether a recommendation is a novel possibility for a user [6].

Also, in this paper, we propose a technique, which is a hybrid of collaborative and content-based approaches, so that its recommendations have a high level of novelty and diversity. Also, to reach advantages of memory-based and model-based approaches, we propose a technique, which is a hybrid of them for collaborative and content-based parts.

The remainder of this paper is organized as follows. In the next section, the proposed dynamic framework for implementing a recommender system is presented. Section 3 describes the dynamic fuzzy clustering algorithm considered for our framework. Section 4 describes the recommender algorithm considered for our system. Experiments with performance evaluation are given in Section 5. The paper concludes in Section 6.

2 The Proposed Dynamic Recommender System Framework

Figure 1 illustrates overall framework which consists of three major components: *task manager*, *recommender system* and *databases* [5]. Task manager includes the user interface which receives the request from the user and decides the appropriate actions to take. The request in the scope of e-commerce includes making the first purchase and become a new customer, purchasing more products, canceling orders, etc. *Recommender system* is composed of *the dynamic fuzzy clustering* and *recommendation engine* modules. The dynamic fuzzy clustering is based on the *Relational Fuzzy Subtractive Clustering (RFSC) algorithm* [18] and the recommendation engine relies on a new hybrid algorithm that will be completely described subsequently. *The database components* consist of *customer profiles*, *item profiles*, *ratings information* and *cluster structures*. Customer profile includes various user characteristics, such as age, gender, income, marital status, etc. Item profile is defined with a set of characteristics; For example, in a movie recommendation application, each item (movie) can be represented by its ID, title, genre, director, year of release, leading actors, etc. Furthermore, ratings information component contains ratings which are assigned to items by customers. In our system, users are clustered based on their preferences and items are clustered based on their contents. The structures of these clusters are stored in the cluster structures component that is dynamically updated by the dynamic fuzzy clustering module over time.

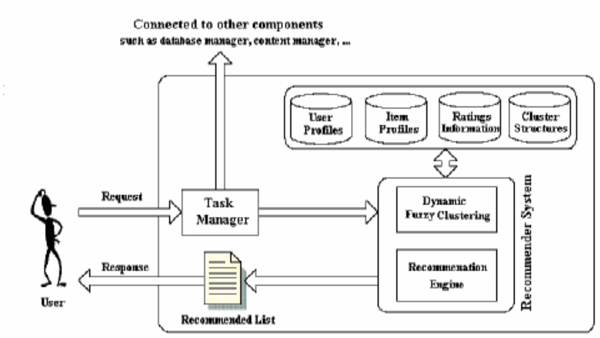


Figure 1. The proposed dynamic recommender system framework

3 Dynamic Fuzzy Clustering

In this work, we use *Relational Fuzzy Subtractive Clustering (RFSC)* [18] which is a highly scalable technique for extracting clusters. 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 [18].

We apply the RFSC algorithm to group the items based on the item contents, such as movie genre, director, actor, actress and even scriptwriter and to group the users based on their interests. Here, we use the cosine similarity measure [2] to measure the similarity among items and the Pearson correlation coefficient to measure the similarity among users [11, 15].

Item profiles or ratings information grow dynamically. Though the clusters obtained from them manifest the groups among items and the interest and trends among the users accessing the system at the time the clustering was applied, the interests and needs of users change dynamically over time. Hence we require some cluster maintenance scheme by which these changing trends and patterns can be captured without having to apply frequently this relatively expensive operation of reclustering of large volume of old and new data together [17].

Cluster maintenance is not a complete alternative to reclustering. By its very definition, cluster maintenance tries to incorporate newly arrived data into the existing model, while maintaining the clusters created from the originally given set of data. It can therefore at best be a close approximation to clustering of the complete data, old plus new. Thus reclustering will always give more accurate results and will be required to be done periodically. Cluster maintenance however enables us to adapt to the dynamic and changing environment in a much less expensive manner in terms of computation times and resources and also enables consecutive maintenance even after reclustering. Thus an optimal combination of full data clustering and cluster maintenance is ideally suited for dynamic environments [17]. In this section, based on [4], we describe our dynamic RFSC algorithm that is an extension of the RFSC algorithm and forms part of our usage profile maintenance scheme. Within our maintenance scheme, we also present a measure that enables us to determine the stage at which a complete reclustering is required.

Let us assume that we have a set of N_U objects and C clusters. Also u ($N_{OBJ} \times C$) is the membership matrix which contains the membership values of each object to each of the C clusters. To begin with, these C clusters and the matrix u are obtained using RFSC algorithm. Subsequently, these are updated using dynamic RFSC until reclustering of the complete dataset is required. These cycles of dynamic RFSC and reclustering make up the maintenance scheme.

Let D be the minimum distance among cluster centers, i.e. $D = \min(D_{C_i,C_j})\ \forall i, j \in [1, C]$. When a new object x_k is inserted, its distances with the existing cluster centers will be measured ($d_{i,k}\ \forall i \in [1..C]$), then its membership degrees to the existing clusters will be calculated and finally it will be inserted into the cluster structure. If $|u_{i,k} - \frac{1}{C}| \le \alpha$ and $d_{i,k} > \frac{D}{2}$ for any $i \in [1..C]$, then this new object is not well classified by the model (α can be determined context-dependent by a system administrator). If the percent of the new objects not well classified by the model is above a predefined threshold β specified by a system administrator, then a complete reclustering has to be performed.

4 Recommendation Engine

A recommender system must be able to recommend the items that have not been yet rated by any user so that its recommendations have high novelty and diversity. Collaborative algorithms and recommender algorithms involving collaborative filtering in their final stage (such as "collaboration via content") can not recommend such items because none of them can make a prediction about a given item unless some users have rated it, therefore we have to use a content-based algorithm but a content-based algorithm suffers the problem of rating sparsity [1]. By using solely this algorithm, we would be restricted to propose only similar items, bore the user and make him/her gradually abandon the system. Moreover, no elements are included that could characterize the quality of the item. To eliminate this disadvantage, the opinions of other users are considered; therefore, we use a new hybrid approach.

Figure 2 presents a flow diagram for our recommendation process. In proposed approach, at first, the collaborative part predicts ratings for a user and then a content-based profile is generated for the user using his/her previous ratings and these new predicted ratings and finally content-based part recommends items using this profile. Where both a predicted rating from collaborative filtering and an actual user rating are available for an item, the content-based part uses the actual rating because it more accurately indicates the user's feelings about the item. By this approach, the items that have not been yet rated by any user can be recommended.

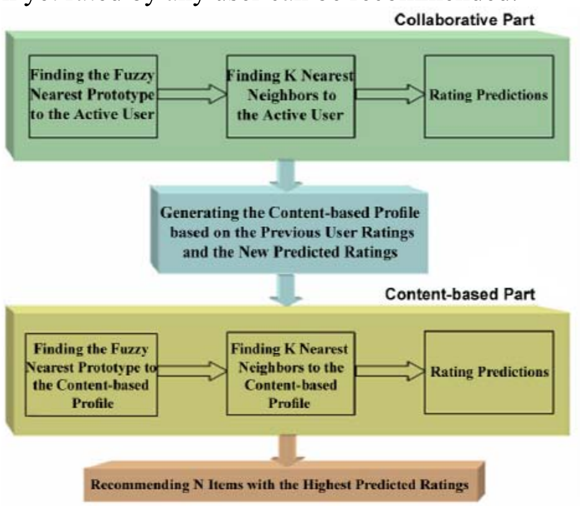


Figure 2. The proposed recommender algorithm

4.1 Collaborative Part

Based on [16], we devise a CF technique whose accuracy is comparable to that of memory-based CF approach with scalability comparable to model-based approach.

We apply the RFSC algorithm to group the users based on their interests. Here, we use the Pearson correlation coefficient to measure the similarity among users [11, 15]. We use two important properties of model learnt from the RFSC algorithm, cluster centers and membership values. RFSC computes clusters whose centers are actual users. 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 user usr_t to cluster i is proportional to its distance or dissimilarity from the cluster center Z_i. The membership values of all the users are stored in matrix u ($N_{USR} \times C$) that there is an index on any cluster in the related database component to reduce search time. For an active user usr_a, we first find the fuzzy nearest prototype, i.e. the cluster p to which the membership u_{pa} is maximum. Now past like-minded users of usr_a will have their memberships close to u_{pa}, thus simplifying the computation of K neighbors enormously. For these K-neighbors, we predict ratings for only those items which have not been yet rated by usr_a.

In the above algorithm, instead of finding the fuzzy nearest prototype, we could find fuzzy M-nearest prototype. The remaining algorithm can be carried out for each cluster in the same way, and at the end, predicted ratings from each cluster can be combined. Also if an item is recommended from more than one cluster, then we consider maximum predicted rating 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.

4.2 Content-based Part

Again, based on [16], we devise a content-based technique whose accuracy is comparable to that of memory-based approach with scalability comparable to model-based approach.

We apply the RFSC algorithm to group the items based on the item contents. Here, we use the cosine similarity measure [2] to measure the similarity among items. 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 item itm_t to cluster i is proportional to its distance or dissimilarity from the cluster center Z_i. The membership values of all the items are stored in matrix u ($N_{ITM} \times C$) that there is an index on any cluster in the related database component to reduce search time. For an active user usr_a, we first make its content-based profile (cbp) using an averaging approach, such as Rocchio algorithm [12], as an "average" vector from individual content vectors. Then we find the fuzzy nearest prototype, i.e. the cluster p to which the membership $u_{p,cbp}$ is maximum. Now the similar items of cbp will have their memberships close to $u_{p,cbp}$, thus simplifying the computation of K neighbors enormously. For these K-neighbors, we predict ratings for only those items which have not been yet rated by usr_a. If the number of desired recommendations is N, then the top-N items, sorted in the order of their predicted ratings, will be presented.

Again, in the above algorithm, instead of finding the fuzzy nearest prototype, we could find fuzzy M-nearest prototype. Like collaborative part, at the end, predicted ratings from each cluster can be combined accordingly.

5 Experimental Evaluations

We wanted to compare our technique with three others:

1) Collaborative technique used in isolation
2) Content-based technique used in isolation
3) "Collaborative-via-content" technique.

5.1 Dataset

The implementation of our approach was based on the MovieLens data set that was collected by the GroupLens Research Project at the University of Minnesota through the MovieLens web site [7].

The characteristics of this set are:

1) 1000,209 evaluations (in a scale from 1 to 5) of 3,592 films by 6,040 users.
2) Each user has evaluated at least 20 films.
3) The types to which films belong are found to be in accordance with the types in the Internet Movies Database (www.imdb.com).
4) Simple demographic information for the users (sex, profession, age) is available. However, we didn't use this information, since its contribution is not big enough to justify the increase in system complexity.

Moreover, we retrieved the synopsis of each movie and the contributors (director, actors and script writers) automatically by using a parser and the URLs provided by MovieLens.

The user evaluates films that he/she has seen in a scale of five degrees (5: masterpiece to 1: bad film). Films worth suggestion are considered those that receive grades 4-5, while those that receive 1-3 are rejected.

By the term "content of film" we refer to a set of elements that can determine verbally the parameters of the film. An important parameter is the kind of a film. Furthermore, the contributors of a film are taken into account, namely directors, script writers and actors.

Starting from the initial data set five distinct splits of training and test data were generated. The final result is the average of results obtained from these five distinct splits.

5.2 Evaluation Metrics

Numerous metrics are available for evaluating recommender systems [6]. Accuracy metrics evaluate how well a system can predict a rating for a specified item, which is a key measure of a recommender system's success. We chose to use *precision* and *recall* in our testing because they're popular, well-established metrics from the information retrieval community. Precision measures the probability that the system's selected films will be relevant to the user, while recall measures the probability that the system will select the entire set of relevant films.

Nevertheless, accuracy alone isn't sufficient proof of a recommender's usefulness. For example, a recommender might be highly accurate but produce rating predictions for only a small number of items. Therefore, we also use *coverage metrics* to indicate the number of films our system can produce predictions about. We used two types of coverage in our testing [6]:

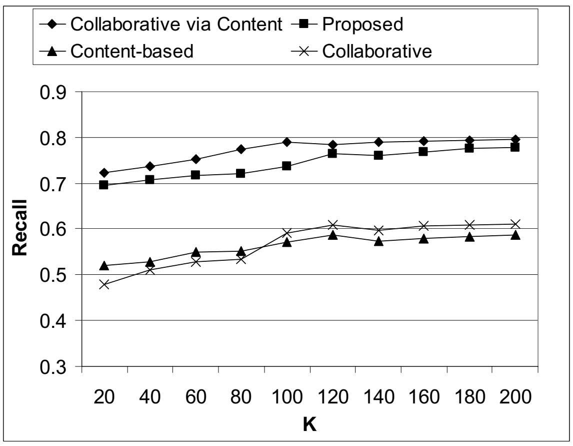
1) *Catalog coverage* shows the percentage of films in the database for which the system ever generates predictions. We use it as a measure of recommendation diversity.

2) *Prediction coverage* measures the average number of films for which the system can produce predictions for each user.

5.3 Comparing Recommendation Accuracy of the Approaches

For this experiment, we fixed the value N to 10 and the value M (number of the fuzzy nearest prototypes) to 2. Figure 3 shows the comparison of recommendation accuracy of different approaches of collaborative, content-based, "collaborative via content" and the proposed hybrid.

It can be seen that the size of the neighborhood has a significant impact on the recommendation accuracy. The quality increases with K and remains almost constant after some point. The effectiveness of our hybrid and "collaborative via content" approaches is much better than collaborative and content-based approaches because these hybrid approaches overcome the problem of rating sparsity well. Furthermore, the precision and recall of the proposed approach is slightly less than those of "collaborative via content" approach because the later operates in a finer granularity level than the former. As a result, employing our approach is justifiable.



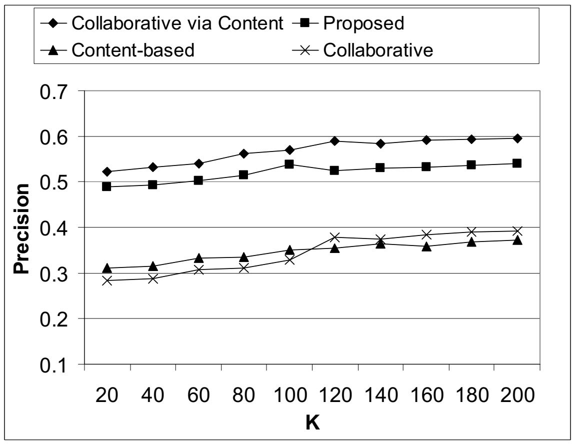
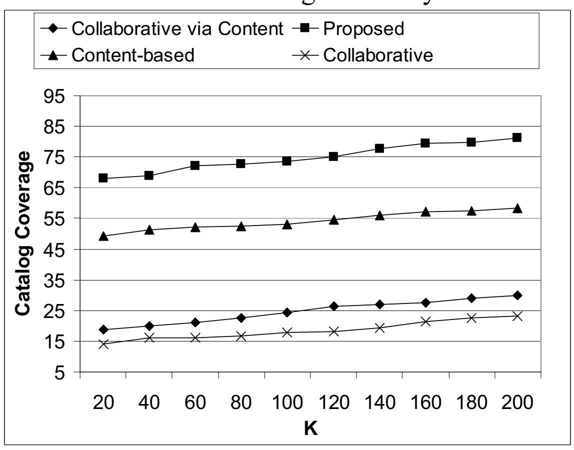


Figure 3. Impact of neighborhood size K on recommendation accuracy

5.4 Comparing recommendation diversity of the approaches

Figure 4 shows the comparison of different approaches in terms of recommendation diversity by keeping N constant at 10 and varying K.

The coverage (catalog and prediction) of our hybrid and content-based approaches is much better than collaborative and "collaborative via content" approaches because collaborative algorithms and recommender algorithms involving collaborative filtering in their final stage (such as "collaboration via content") can not make a prediction about a given movie unless some users have rated it. Our technique's coverage (catalog and prediction) is higher than content-based technique because the former relatively overcome the problem of rating sparsity. As a result, our technique's recommendations have a high diversity level.



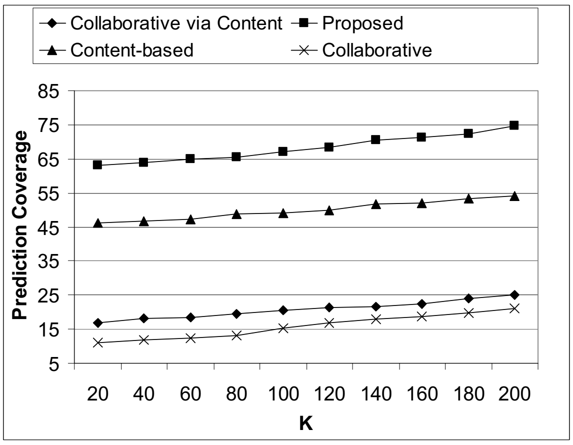


Figure 4. Impact of neighborhood size K on recommendation diversity

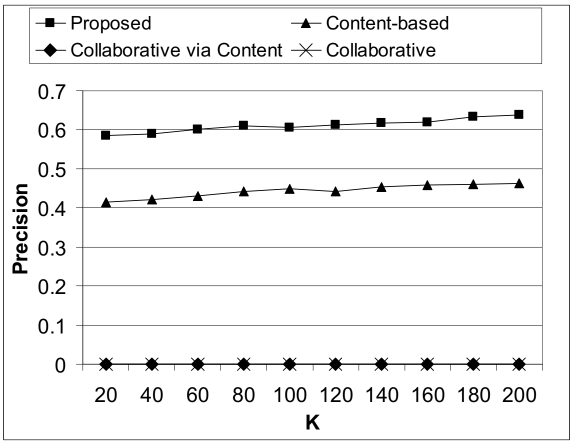
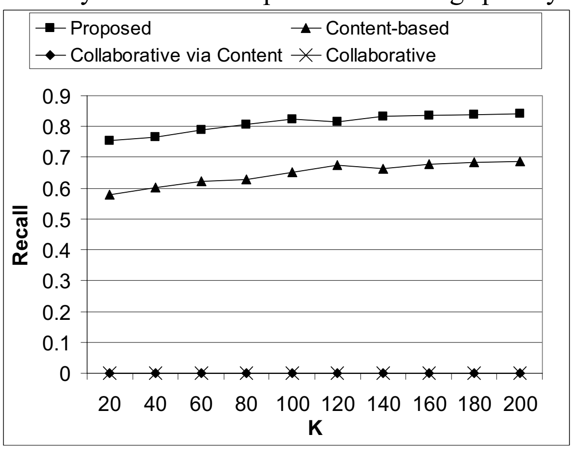


Figure 5. Impact of neighborhood size K on recommendation novelty

5.5 Comparing Recommendation Novelty of the Approaches

For this experiment, first, we removed all ratings for a set of films (400 movies), thus simulating new movies that users hadn't yet rated. Then we used the remaining rating information for each user and attempted to regenerate predictions for this film set. Figure 5 shows the comparison of different approaches in terms of recommendation novelty by keeping N constant at 10 and varying K.
The recall and precision are zero for collaborative and "collaborative via content" techniques because none of them can make a prediction about a given movie unless some users have rated it. Again, our technique's accuracy (recall and precision) is higher than content-based technique because the former relatively overcome the problem of rating sparsity.



5.6. Impact of Applying the Dynamic RFSC Algorithm

The main goal of adapting the model to the dynamic changes in the environment is to increase the relevance and accuracy of recommendations to the user. We now discuss the results from some of the experiments performed to determine the recommendation quality when maintenance algorithm was used. we compare recommendation quality for the following three techniques:
1) Our hybrid technique with dynamic RFSC
2) Our hybrid technique with reclustering
3) Our hybrid technique without maintenance algorithm
For this experiment, we kept the parameter K-nearest neighbors, K=100; fuzzy M-nearest prototypes, M=2; and varied the parameter N from 5 to 50. Moreover, after generating recommendation for any 50 users, we considered their actual ratings on items in the test set as their implicit feedback and we applied dynamic RFSC for these new ratings and for comparison, applied reclustering of the entire ratings, old plus new. Here, we chose $\alpha = 0.05$ and $\beta = 80\%$.
It can be seen that, the precision and recall of the system with dynamic RFSC are nearly between those of the system with reclustering and of the system without maintenance algorithm. In Figure 7, we show the comparison of efficiency in terms of online recommendation time. As can be seen, the time taken for the reclustering approach is much higher than the other approaches. Also, the time

taken for the dynamic RFSC approach is slightly higher than the approach without maintenance algorithm. Therefore, the system is able to adapt to the dynamic and changing environment in a much less expensive manner in terms of computation times and resources.

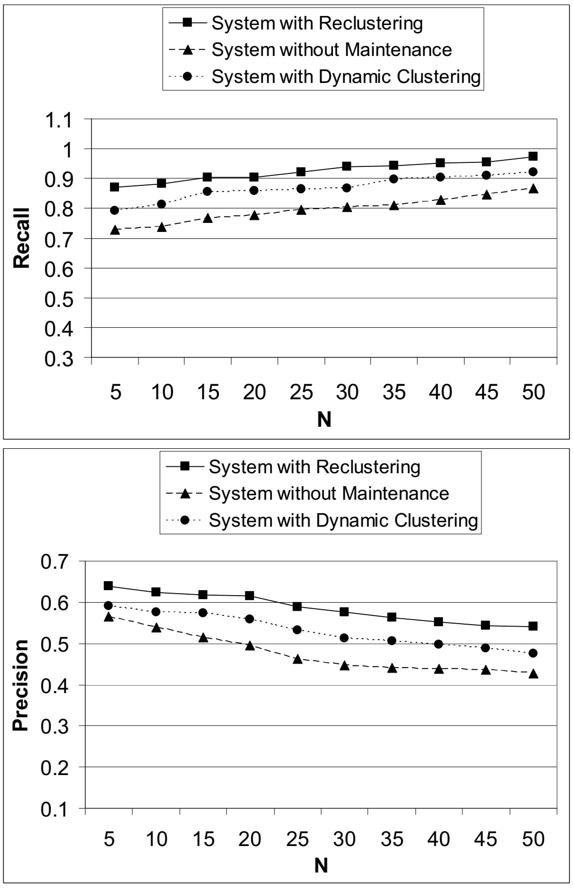


Figure 6. Impact of applying the dynamic RFSC algorithm on recommendation accuracy

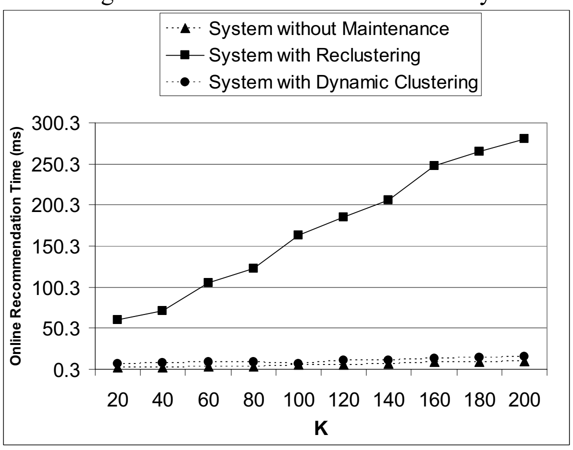


Figure 7. Impact of applying the dynamic RFSC algorithm on efficiency

6 Conclusions and Future Works

In this paper, a new hybrid system was proposed for combining collaborative and content-based approaches that resolves some limitations of them. Specially, by proposed system, the novelty and diversity of recommendations improve remarkably. Furthermore, the precision and recall of the proposed system is slightly less than those of the best existing hybrid system (collaborative via content) so that employing this system is justifiable. In proposed system, at first, the collaborative part predicts ratings for a user and then a content-based profile is generated for the user using his/her previous ratings and these new predicted ratings and finally content-based part recommends items using this profile. By this approach, the items that have not been yet rated by any user can be recommended. Collaborative and content-based systems utilized by this work, use a hybrid method based on fuzzy clustering model (fuzzy subtractive clustering) that combines model and memory-based approaches so that its precision is comparable with the precision of the memory-based approach and its scalability is comparable with the scalability of the model-based approach.

To generate useful recommendations to satisfy the user, a recommender system not only must be able to handle a large number of users and items bus also must adapt itself to changing interests and needs. In this work, a dynamic fuzzy clustering algorithm was proposed in which a measure is presented to determine the stage at which a complete reclustering is required. By applying this algorithm, the precision and recall of the system, are nearly between those of the system applying reclustering and of the system without maintenance algorithm; therefore, the system is able to adapt to the dynamic and changing environment in a much less expensive manner in terms of computation times and resources.

We believe we can further improve our system's prediction accuracy by improving the content-based filtering algorithm. We will investigate methods to create more complex groupings of actors, directors, and genres, rather than treating each as an independent entity.

We want to test machine learning methods for dynamically altering the component weightings within the content-based filtering algorithm. For example, a highly rated director might have more influence on a film recommendation than a highly rated actor.
There are users of recommender systems whose goal is to explicitly influence others into viewing or purchasing particular items. For example, advocates of particular movie genres (or movie studios) will frequently rate movies high on the MovieLens web site right before the movie is released to try and push others to go and see the movie. Therefore we may want to find ways that system can prevent this task.

7 References

[1] G. Adomavicius and A. Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734-749, June 2005.
[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.
[3] J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", *In Proc. of UAI-98*, pp. 43-52, 1998.
[4] F. Crespo and R. Weber, "A Methodology for Dynamic Data Mining based on Fuzzy Clustering", *Fuzzy Sets and Systems*, Vol. 150, Iss. 2, pp. 267-284, 2005.
[5] C. Haruechaiyasak, C. Tipnoe, S. Kongyoung, C. Damrongrat, and N. Angkawattanawit, "A Dynamic Framework for Maintaining Customer Profiles in E-Commerce Recommender Systems", *Proc. Int'l Conf. e-Technology, e-Commerce and e-Service (EEE'05)*, pp. 768-771, 2005.
[6] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating Collaborative Filtering Recommender Systems", *ACM Trans. Information Systems*, Vol. 22, No. 1, pp. 5-53, 2004.
[7] http://movielens.umn.edu
[8] O. Nasraoui, "World Wide Web Personalization", *Encyclopedia of Data Mining and Data Warehousing*, J. Wang, ed., Idea Group, 2005.
[9] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, , "Collaborative Filtering by Personality Diagnosis, a Hybrid Memory- and Model-based Approach", *In Proc. of UAI-2000*, Stanford, CA, pp. 473-480, 2000.
[10] D. Pierrakos, G. Paliouras, C. Papatheodorou, C. Spyropoulos, and C. Koinotites, "A Web Usage Mining Tool for Personalization", *In Proc. of the Pan-Hellenic Conference on Human Computer Interaction*, Patras, December 2001.
[11] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", *Proc. 1994 Computer Supported Cooperative Work Conf.*, 1994.
[12] J.J. Rocchio, "Relevance Feedback in Information Retrieval", *SMART Retrieval System - Experiments in Automatic Document Processing*, G. Salton, ed., Chapter 14, Prentice Hall, 1971.
[13] B. M. Sarwar, G. Karypis, J.A. Konstan, and J. Riedl, "Analysis of Recommender Algorithms for E-Commerce", In *Proc. of the 2nd ACM E-commerce Conference*, Minnesota, USA, 2000.
[14] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-Commerce Recommendation Applications", *Data Mining and Knowledge Discovery*, Vol. 5, Nos. 1/2, pp. 115–153, 2001.
[15] U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth'", *Proc. Conf. Human Factors in Computing Systems*, 1995.
[16] B.S. Suryavanshi, N. Shiri, and S.P. Mudur, "A Fuzzy Hybrid Collaborative Filtering Technique for Web Personalization", *Proc. 3rd Workshop Intelligent Techniques for Web Personalization (ITWP'05)*, Edinburgh, Scotland, Aug. 2005.
[17] B.S. Suryavanshi, N. Shiri, and S.P. Mudur, "Incremental Relational Fuzzy Subtractive Clustering for Dynamic Web Usage Profiling", *Proc. 3rd Workshop Intelligent Techniques for Web Personalization (ITWP'05)*, Edinburgh, Scotland, Aug. 2005.
[18] B.S. Suryavanshi, N. Shiri, and S.P. Mudur, "An Efficient Technique for Mining Usage Profiles using Relational Fuzzy Subtractive Clustering", *Proc. IEEE Int'l Workshop Challenges in Web Information Retrieval and Integration (WIRI' 05)*, Tokyo, Japan, April 8-9, 2005.