International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:3, 2011

A Hybrid Recommender System based on Collaborative Filtering and Cloud Model

Chein-Shung Hwang and Ruei-Siang Fong

***Abstract*—**User-based Collaborative filtering (CF), one of the most prevailing and efficient recommendation techniques, provides personalized recommendations to users based on the opinions of other users. Although the CF technique has been successfully applied in various applications, it suffers from serious sparsity problems. The cloud-model approach addresses the sparsity problems by constructing the user's global preference represented by a cloud eigenvector. The user-based CF approach works well with dense datasets while the cloud-model CF approach has a greater performance when the dataset is sparse. In this paper, we present a hybrid approach that integrates the predictions from both the user-based CF and the cloud-model CF approaches. The experimental results show that the proposed hybrid approach can ameliorate the sparsity problem and provide an improved prediction quality.

***Keywords*—**Cloud model, Collaborative filtering, Hybrid recommender system

I. Introduction

WITH the rapid development of information technology, more and more information is created, distributed and accessed over the internet. However, the amount of available information is so overwhelming that people cannot readily digest it. This scenario is commonly referred to as the "information overload" problem. Researchers have developed many different techniques to address the information overload problem. Examples of such techniques include web search engines, intelligent agents, recommender systems [1]-[5]. Among them, the most widely used technique is the search engines that help people locate their desired information using keywords. However, traditional retrieval strategies provide limited personalized services and yield increasing poor results due to the rapid increases of internet pages. Intelligent agents are proactive, customizable online search, retrieval and notification softwares that act as a filter that allows the information of particular interest or relevance to users to get through and blocks off the flow of useless or irrelevant information. Today, intelligent agents are deployed in different settings to reduce work and information overload, such as e-mail organization, news filtering, personal assistance, meeting scheduling and many others [6]-[9]. However, extra efforts from users are required to specify his/her interests or answer additional questions [10].

Recommender systems solve the information overload problem by helping users discover and evaluate items of interest. Collaborative filtering (CF) [11]-[12] is one of the most prevailing and efficient techniques used in recommender systems. The main idea behind CF algorithms is to automate the process of "word-of-mouth" by which people recommend items to one another. For each user, CF algorithms use historical information to identify a neighborhood of people who have shown similar behavior in the past and then predict the interest of new items by analyzing the neighborhood. Although the CF technique has been successfully applied in various applications, it suffers from two major problems: sparsity and scalability [13]-[14]. The sparsity problem occurs when available ratings data is rare and insufficient for identifying the similar neighbors. This problem is often very significant when the system is in its early stages. Sparsity of ratings data is the major reason causing poor recommendation quality. Several methods have been proposed to deal with the sparsity problem. For example, Billsus and Pazzani [15] used Singular Value Decomposition (SVD) to reduce the dimensions of ratings data by removing unrepresentative users or items. Ziegler et al. [16] incorporated taxonomic information into the CF algorithm for facilitating the inference of profile similarity in the sparse ratings data. Piccart et al. [17] addressed the sparsity problem by switching the ratings to a probabilistic representation of the rankings. Instead of using ratings data, Hwang and Chen [18] permitted the exploration of transitive user similarities based on the trust inference in a social network.

CF recommender systems also have to make recommendations real-time. However, the CF algorithm requires computations that are expensive and grown nonlinearly with both the number of users and items. The poor scalability of the CF algorithm makes it inefficient for a real-time implementation. Several attempts have been made to deal with the scalability problems. Sarwar et al. [19] developed an incremental SVD CF algorithm that used the folding-in projection technique to derive the SVD decomposition when a set of new ratings is added to the database. Linden et al. [20] alleviated the scalability problem by calculating the relationships between items rather than the relationships between users. Hwang and Tsai [21] used a model-based approach by seeking users for recommendation within smaller and highly similar clusters.

Recently, several researchers [22]-[23] have claimed the effectiveness of incorporating the cloud model into the CF process. The cloud-model CF approach addresses the sparsity

Chein-Shung Hwang is with the Department of Information Management, Chinese Culture University, 55, Hwa-Kang Road, Yang-Ming-Shan, Taipei, Taiwan (e-mail: cshwang@faculty.pccu.edu.tw).

Ruei-Siang Fong is with the Department of Information Management, Chinese Culture University, 55, Hwa-Kang Road, Yang-Ming-Shan, Taipei, Taiwan (e-mail: zak@goldaries.com).

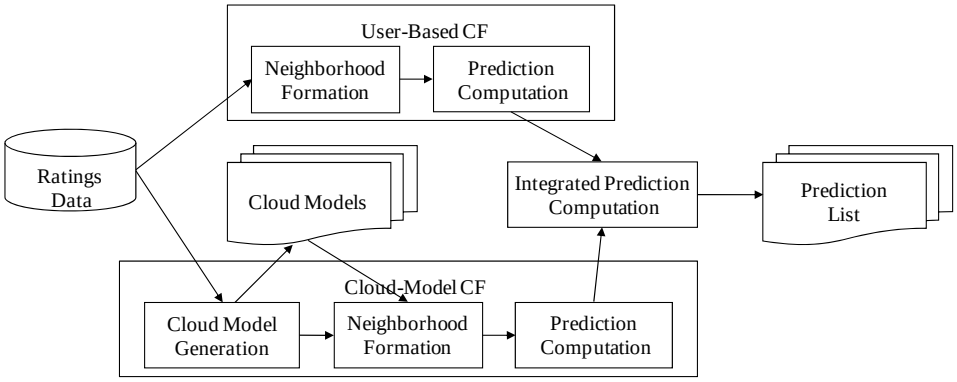

Fig. 1 System architecture

and the scalability problems by constructing the user's global preference represented by a cloud eigenvector. This method can reduce the dimensionality of data and avoid the strict matching of attributes in similarity computation [22]. However, we argue that the use of the global preference for similarity computation may lose some information about individual ratings and accordingly result in a poor recommendation quality. The example in Table I illustrates a fragment of movie rating data for users A, B, and C, over ten items. Suppose we attempt to predict the rating of user A for item 6 using the 1-NN method. The user-based CF approach will choose user C as the nearest neighbor and provide a rating of 5. In contrast, the cloud-mode CF approach will select user B and predict a rating of 2, even there are no co-rated items between user A and user B. The difference of these two predictions is due to different similarity measures. Traditional similarity measures cannot calculate the similarity of two users if they have not rated any identical item. The cloud-model CF approach solves this problem by providing a global similarity measure. Although the cloud model can improve the predictive coverage of the CF algorithm, it may degrade the predictive accuracy.

In this paper, we present a hybrid approach by taking advantages from both the user-based CF and the cloud-model CF approaches. The user-based CF approach works well with dense datasets while the cloud-model CF approach has a greater performance when the dataset is sparse. In particular we will define a unified prediction method that integrates the predictions from these two approaches based on the number of users who have made a contribution to the predictions. The performance of several different similarity measures on the predictive coverage and accuracy for different approaches will be evaluated and compared.

TABLE I
A FRAGMENT OF A MOVIE RATING MATRIX

Item / *User*	1	2	3	4	5	6	7	8	9	10
A	4	3	5	1	2	?	-	-	-	-
B	-	-	-	-	-	2	1	3	5	2
C	4	3	5	1	2	5	-	-	-	-

II. HYBRID COLLABORATIVE FILTERING SYSTEM

The proposed hybrid CF system integrates the predictions from both the user-based CF and the cloud-model CF subsystems. The overall system can be viewed as a blackbox which takes as input the ratings matrix and produces, as output, a prediction list. The ratings matrix R contains the rating values $r_{u,i}$, standing for the rating of user u for item i. The user-based CF subsystem uses the traditional CF algorithm to provide predictions based on individual ratings data. The cloud-model CF subsystem first transforms the ratings data into the user global preferences and then performs a neighborhood-based algorithm to generate predictions. Finally, the system combines the predicted ratings from each subsystem and generates a list of predictions.

A. User-Based CF Subsystem

The user-based CF subsystem implements a nearest neighbor algorithm that calculates the similarity between two users and produces a prediction for the user by taking the weighted average of all the ratings.

1) Neighborhood Formation

The neighborhood formation (NF) module finds the most similar neighbors for an active user based on the similarity measures between them. Two most commonly used approaches for computing the similarity between users are: Pearson correlation coefficient and cosine similarity. Pearson correlation measures the extent to which a linear relationship exists between two variables. The correlation measure can range from -1 to 1, with -1 indicating a perfect negative linear correlation, 1 indicating a perfect positive linear correlation and 0 indicating no linear correlation between two variables. In the user-based CF algorithm, Pearson correlation coefficient is often used to define the similarity based on the common ratings of two users:

$$sim(u,v) = corr(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}} \quad (1)$$

where I_{uv} is the set of items rated by both user u and v, $r_{u,i}$ is the

rating for item *i* given by user *u*, and $\bar{r}_u$ is the average rating given by user *u*. Herlocker et al. [26] have suggested that taking the number of co-rated items into account while computing the similarity can provide a better predictive accuracy. In order to achieve a better implementation, we define a significance weighting scheme that weights the Pearson correlation using a smooth devaluation as

$$sim(u,v) = corr(u,v) \times \frac{m}{1+m}, \quad (2)$$

where *m* is the number of co-rated items between user *u* and *v*. The cosine similarity approach considers each user's set of ratings as a vector and uses the cosine of the angle between two users' ratings vectors as a measure of their similarity. The cosine measure has value in the range [0, 1], with 1 indicating the perfect match of two users and 0 indicating an complete difference. The similarity of two users *u* and *v* based on the cosine measure is expressed as:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_u} r_{u,i}^2} \sqrt{\sum_{i \in I_v} r_{v,i}^2}}, \quad (3)$$

where I_u and I_v denote the set of items rated by user *u* and *v*, respectively. The cosine similarity does not take into account the difference in rating scales between different users. To address this drawback, adjusted cosine similarity, a variant of cosine similarity, is used by subtracting the corresponding user average from each co-rated pair. The adjusted cosine similarity is computed as:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_v} (r_{v,i} - \bar{r}_v)^2}} \quad (4)$$

The NF forms a neighborhood for the active user by selecting *k* users that have the highest similarities.

2) Prediction Computation

The goal of the PC module is to predict the ratings of an unrated item for the active user. After selecting the *k* most similar users, their ratings and the similarity measures as well as the overlapping degree are used to generate the prediction as follows.

$$p_{u,i}^{CF} = \bar{r}_u + \frac{\sum_{v \in N(u)} (r_{v,i} - \bar{r}_v) \times sim(u,v)}{\sum_{v \in N(u)} sim(u,v)} \quad (5)$$

where $N(u)$ is the *k* most similar users to active user *u* and $\bar{r}_u$ and $\bar{r}_v$ are the average rating of user *u* and *v*, respectively. The value of rating $r_{v,i}$ is weighted by the similarity of user *v* to user *u*; the more similar the two users are, the more weight $r_{v,i}$ will have in the computation of the predicted rating $p_{u,i}^{CF}$.

B. Cloud-Model CF Subsystem

The cloud-model CF system consists of three modules. The cloud model generation (CMG) module computes the cloud model for each user based on the ratings data. The cloud model represents the user's global preference. The neighborhood formation (NF) module finds the most similar neighbors for an active user based on the similarity measure of the corresponding cloud models. The prediction computation (PC) module computes the predicted ratings of unrated items for the active user.

1) Cloud Model Generation

The cloud model, proposed by De-yi Li [24], is a model of uncertainty transition between qualitative concept and its quantitative representation. As the ratings data collected by the CF systems is based on user's perceptions, opinions and tastes, which are subjective, imprecise, and vague [25], cloud model seems to be an appropriate paradigm to handle the uncertainty and fuzziness on user preference.

The goal of CMG module is to construct the global preference for each user. The global preference is represented by the cloud model. For each user, the CMG module reads his ratings data and computes the corresponding cloud model in terms of a triple of three digital characteristics $\overrightarrow{V} = (Ex, En, He)$. The expected value *Ex* represents the typical value of user ratings, that is, the average of user ratings. The entropy *En* represents the uncertainty distribution of user preference, which is measured by the deviation degree from the average rating. The hyper-entropy *He* is a measure of the uncertainty of the entropy *En*, which is measured by the deviation degree from the normal distribution. Given a set of ratings data for a user *u*, $r_u = (r_{u,1}, r_{u,2}, \ldots, r_{u,n})$, the three characteristics can be defined as [22]:

$$\begin{aligned} Ex &= \frac{1}{n}\sum_{i=1}^{n} r_{u,i} \\ En &= \sqrt{\frac{\pi}{2}} \times \frac{1}{n}\sum_{i=1}^{n} \left| r_{u,i} - Ex \right| \\ He &= \sqrt{S^2 - \frac{1}{3}En^2}, \quad \text{where } S = \frac{1}{n-1}\sum_{i=1}^{n} (r_{u,i} - Ex)^2 \end{aligned} \quad (6)$$

2) Neighborhood Formation

An important step of the neighborhood-based algorithm is to find the neighbors (similar users) for an active user. The neighbors are selected based on the cloud model similarities between the active user and the users in the database. Zhang et al. [22] proposed a likeness similarity method based on cloud model using the cosine measure. Given two cloud models in terms of the characteristic vectors $\overrightarrow{V_u} = (Ex_u, En_u, He_u)$ and $\overrightarrow{V_v} = (Ex_v, En_v, He_v)$, the similarity between them are defined as

$$sim(u,v) = \cos(\overrightarrow{V_u}, \overrightarrow{V_v}) = \frac{Ex_u Ex_v + En_u En_v + He_u He_v}{\sqrt{Ex_u^2 + En_u^2 + He_u^2} \sqrt{Ex_v^2 + En_v^2 + He_v^2}}. \quad (7)$$

As mentioned previously, part of ratings information would be lost when turning them into the global preference. It is possible to overestimate the similarity between two users who have rated very few items in common. To alleviate this problem, we adopt the same weighting scheme as that for Pearson coefficient.

$$sim(u,v) = \cos(\overrightarrow{V_u}, \overrightarrow{V_v}) \times \frac{m}{m+1}, \quad (8)$$

where m is the number of common ratings between user u and v. The proposed weighting scheme prefers the neighbors that have many common ratings with the active user. Neighbors with very few common ratings will be eliminated from the neighborhood, even if there is a very high degree of the global preference similarity to the active user. Finally, the NF forms a neighborhood for the active user by selecting k users that have the highest similarity.

3) Prediction Computation.

The process of the PC module in the cloud-model CF subsystem is similar to that in the user-based CF subsystem. The PC module predicts the ratings of an unrated item for the active user using a weighted average of all similar users who have rated that item.

$$p_{u,i}^{CM} = \bar{r}_u + \frac{\sum_{v \in N(u)} (r_{v,i} - \bar{r}_v) \times sim(u,v)}{\sum_{v \in N(u)} sim(u,v)} \quad (9)$$

C. Integrated Prediction Computation.

After the implementations of the two subsystems, we obtain two predictions for each unrated item. The final step of our system is to create a unified prediction by integrating the predictions of these two subsystems. In previous work [27], Lou et al. developed a unified CF framework based on both local similarity and global similarity. They defined a parameter to determine the extent to which the final prediction relied on each individual prediction. This method required a tedious experimental validation and the performance seemed to be application dependent. Instead we combine the predictions using a fixed ratio that measures the relative size of the number of contributors in each individual prediction. The contributors are those users in the neighborhood who have made predictions for an unrated item for the active user. We believe that a prediction is more reliable if more users are participating in making predictions. The unified prediction is therefore defined as the weighted average of the predictions from the two subsystems.

$$p_{u,i} = \frac{N_{u,i}^{CF}}{N_{u,i}^{CF} + N_{u,i}^{CM}} \times p_{u,i}^{CF} + \frac{N_{u,i}^{CM}}{N_{u,i}^{CF} + N_{u,i}^{CM}} \times p_{u,i}^{CM} \quad (10)$$

where $N_{u,i}^{CF}$ and $N_{u,i}^{CM}$ are the number of contributors who have made predictions for item i of user u in the user-based CF and the cloud-model subsystems, respectively.

III. Experimental Evaluation.

A. Data Set

We use the *movielens* (ML) dataset collected by the GroupLens Research at the University of Minnesota. It contains 100,000 ratings from 943 users for 1,628 movies. Each user has rated at least 20 movies, and each movie has been rated at least once. In our experiments, we use the two pairs of datasets; one pair is ua.base and ua.test and the other pair is ub.base and ub.test. These paired datasets split the original dataset into a training set and a test set with exactly 10 ratings per user in the test set. The training set and its corresponding test set are disjoint. The training set contains 90,570 ratings and the test set contains 9,430 ratings. The training set is used to generate the prediction for each item in the test set. Our proposed systems are evaluated by comparing the predicted ratings with the actual ratings of the test items.

B. Evaluation Metrics

To measure the accuracy of the recommendations we computed the standard Mean Absolute Error between actual ratings and predictions in the test set. MAE is a measure of the deviation of recommendations from their actual ratings. Specifically, given the set of actual/predicted pairs ($r_{u,i}$, $p_{u,i}$) for all the movies rated by user u, the MAE for user u is computed as:

$$MAE_u = \frac{\sum_{i \in I_u} |r_{u,i} - p_{u,i}|}{n(I_u)} \quad (11)$$

where I_u represents the set of items that are rated by user u. The overall MAE is computed by averaging these individual MAEs over all users in the test set.

Another important measure for discriminating between different recommender approaches is *coverage*. Coverage is a measure of percentage that a recommender system can provide predictions. It is impossible to make a prediction for an active user on a certain item when very few users have rated the items or the active user has no correlation with other users. Low coverage is usually due to a small size of neighborhood or a sparse database. To evaluate the effectiveness of different approaches in alleviating the data sparsity problem, we compare their performance in terms of coverage with the increase in sparsity level.

IV. Performance Results

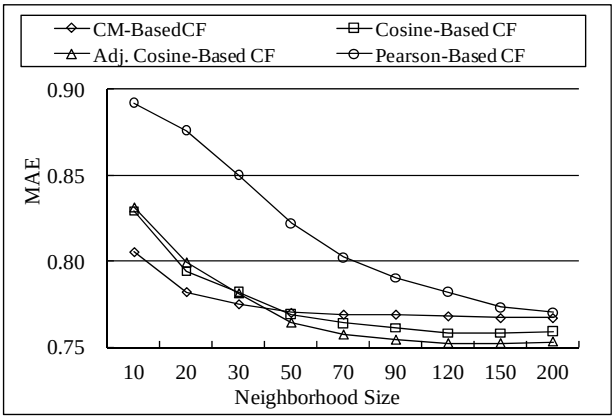

Fig. 2 MAE for different algorithms without significant weighting

Fig. 2 shows the predictive accuracy for different similarity measures without using the significance weighting scheme. The MAE is expressed with respect to the neighborhood size.

In all approaches, the MAE improves as the neighborhood size increases but most of them (except the Pearson-based CF approach) reach a stable performance around 90 neighbors and any further increment makes no better or even worse results. The Pearson-based CF approach performs the worst overall, followed by the cloud-model CF approach. The adjusted cosine-based CF approach and the cosine-based CF approach perform similarly and have the best predictive accuracy in all cases. These results confirm our claims that the use of the global preference may lead to some accuracy degradation.

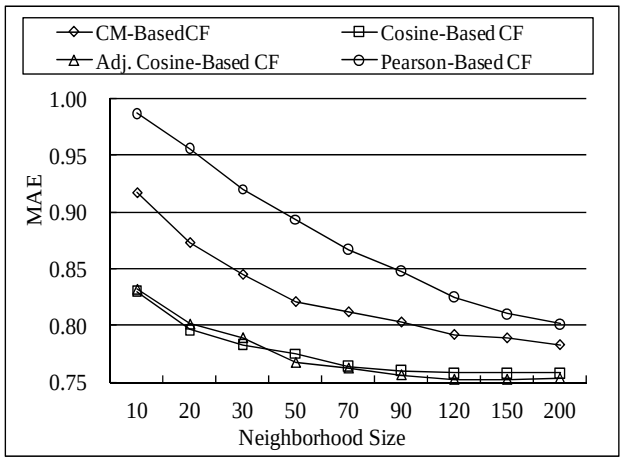

Fig. 3 MAE for different algorithms with significant weighting

Fig. 3 shows the predictive results when taking the common ratings into account. It can be observed that both the cloud-model CF and the Pearson-based CF approaches have a great predictive improvement in a small neighborhood. The improvement becomes less significant when more users are used for predictions. On the other hand, the cosine-based CF and the adjusted cosine-based CF approaches gain less benefit from adding the significant weighting scheme. Overall, the cloud-model CF approach performs no better or worse than other approaches. The adjusted cosine-based CF approach performs the best in a large neighborhood.

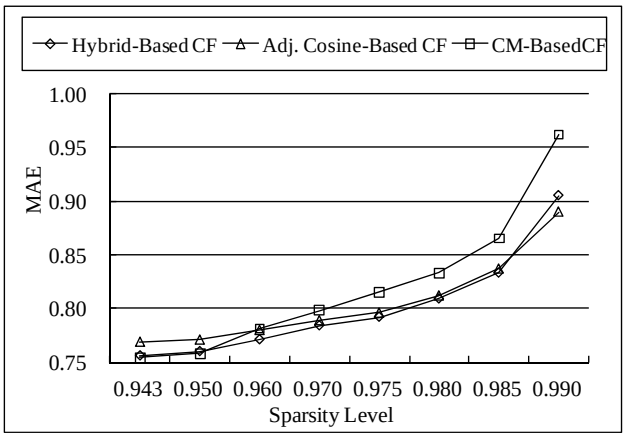

Fig. 4 MAE for different algorithms by varying sparsity levels

In the following experiments, the hybrid-based CF approach (an integration of the adjusted cosine-based CF and the cloud-model CF approaches) is evaluated. To evaluate the performance of different approaches, we fix the neighborhood size to 90 and perform the experiments with different sparsity levels. The sparsity level is measured as the ratio of number of zero entries to the total number of entries in the rating matrix. The sparsity level of the training dataset is about 94.3%. We randomly but evenly remove the ratings from each user in the training dataset to increase the sparsity levels. Fig. 4 shows how the MAE changes with respect to varying sparsity levels. From this figure, we can see the impact of sparse datasets on the predictive accuracy. In all approaches, the predictive accuracy decreases as the sparsity level increases. The hybrid approach leverages the results of individual approaches and performs the best in most cases.

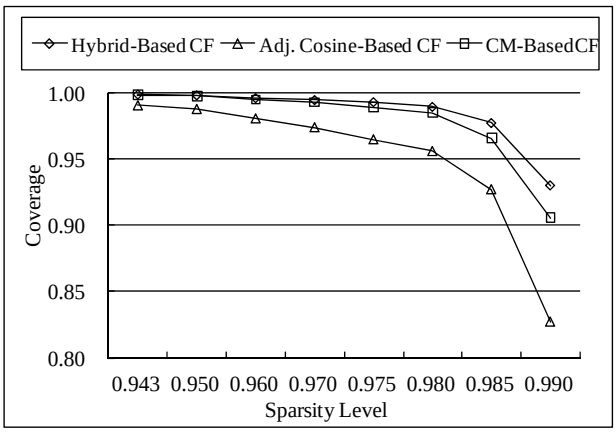

Fig. 5 Coverage for different algorithms by varying sparsity levels

Fig. 5 shows the results of coverage for different CF approaches. The cloud-model CF approach has a better coverage than the adjusted cosine-based CF approach in all cases. This result illustrates the effectiveness of the cloud-model CF approach on a sparse dataset. The performance of all approaches is quite similar in the low sparsity levels. When the level of sparsity increases, the coverage drops gradually. However, the proposed hybrid-based CF approach maintains the best performance in all cases.

V. Conclusions

In this paper, we have proposed a hybrid approach that leverages both the advantages of the user-based CF and the cloud-model CF algorithms to ameliorate the sparsity problem. We compare the performance of different similarity measures and investigate the impact of the significance weighting scheme. Overall, the cloud-model CF approach gains the most benefits from adding the significance weighting. The significance weighing scheme has little effect on the cosine and the adjusted cosine similarity because the problem of similarity overestimation does not occur in these two similarity measures. The proposed hybrid CF approach integrates the predictions from two subsystems using a weighted average scheme.

Experimental results show that the proposed approach can provide improved performance in a sparse dataset. Future work includes a further validation of the proposed system in a real-world application.

REFERENCES

[1] J. Beheshti, "Browsing through public access catalogs," *Information Technology & Libraries*, vol. 11, no. 3, Library and Information Technology Association, pp. 220-228, 1992.

[2] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Clustering user queries of a search engine," in *Proc. 10th International Conference on World Wide Web*, ACM Press, 2001, pp.162-168.

[3] D. Billsus and M. J. Pazzani, "Learning collaborative information filters," in *Proc. ICML*, Morgan Kaufmann Publishers Inc., 1998, pp.46-53.

[4] T.-P. Liang and J.-S. Huang, "A framework for applying intelligent agents to support electronic trading," *Decision Support Systems*, vol. 28, pp.305-317, 2000.

[5] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proc. ACM EC*, ACM Press, 2000, pp.158-167.

[6] R. B. Segal and J. O. Kephart, "MailCat: an intelligent assistant for organizing e-mail," *in Proc. AGENTS '99 Third annual conference on Autonomous Agents*, ACM, 1999, pp.276-282

[7] J.-H. Chiang, Y.-C. Chen, "An intelligent news recommender agent for filtering and categorizing large volumes of text corpus," *International Journal of Intelligent Systems*, vol 19, Issue 3, pp.201–216, March 2004.

[8] G. Czibula, A.-M. Guran, I. G. Czibula, G. S. Cojocar, "IPA - An intelligent personal assistant agent for task performance support," *Intelligent Computer Communication and Processing, 2009, ICCP 2009. IEEE 5th International Conference on.*, pp.31, 2009.

[9] C.-S. Lee and C.-Y. Pan, "An Intelligent Fuzzy Agent for Meeting Scheduling Decision Support System," *Fuzzy Sets and Systems*, vol. 142, no. 3, pp. 467-488, 2004.

[10] A. Birukou, E. Blanzieri, and P. Giorgini. "Implicit: An agent-based recommendation system for web search," in *Proc. 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, ACM Press, 2005, pp. 618–624.

[11] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp.77-87, 1997.

[12] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. "An algorithmic framework for performing collaborative filtering," in *Proc. 1999 Conference on Research and Development in Information Retrieval*, 1999, pp. 230-237.

[13] G. Adomavicius and A. Tuzhilin, "Toward 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, 2005.

[14] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp.331–370, 2002.

[15] D. Billsus and M. Pazzani, "Learning collaborative information filters," in *Proc. 15th International Conference on Machine Learning*, 1998, pp. 46-54.

[16] C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme, "Taxonomy-driven computation of product recommendations," in *Proc. 13th International Conference on Information and Knowledge Management*, 2004, pp. 406–415.

[17] B. Piccart, J. Struyfy, and H. Blockeelz, "Alleviating the sparsity problem in collaborative filtering by using an adapted distance and a graph-based method," in *Proc. SIAM International Conference on Data Mining*, 2010, pp. 189-198.

[18] C.-S. Hwang and Y.-P. Chen, "Using trust in collaborative filtering recommendation," in *Proc. 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2007, pp. 1052-1060.

[19] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental SVD-based algorithms for highly scaleable recommender systems," in *Proc. 5th International Conference on Computer and Information Technology*, 2002, pp. 399-404.

[20] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[21] C.-S. Hwang and P.-J. Tsai, "A collaborative recommender system based on user association clusters," in *Proc. 6th International Conference on Web Information Systems Engineering*, 2005, pp. 463-469.

[22] G. W. Zhang, D. Y. Li, P. Li, J.-C. Kang, and G.-S. Chen, "A collaborative filtering recommendation algorithm based on cloud model," *Journal of Software*, vol. 18, no. 10, pp. 2403-2411, 2007.

[23] R. Luo and G. Yuxi, "Personalized recommendation based on similarity of cloud model," in *Proc. 2nd International Symposium on Knowledge Acquisition and Modeling*, 2009, pp. 356-359.

[24] D. Y. Li, C. Y Liu, Y. Du, and X. Han, "Artificial intelligence with uncertainty," *Journal of Software*, vol. 15, no. 9, pp. 1583-1594, 2004.

[25] K, Palanivel and R, Siavkumar, "Fuzzy multicriteria decision-making approach for collaborative recommender systems," *International Journal of Computer Theory and Engineering*, vol. 2, no. 1, pp. 57-63, 2010.

[26] J. L. Herlocker, J. A. Konstan, A. Borchers, J. Riedl, 1999. "An algorithmic framework for performing collaborative filtering," in *Proc. 22nd International Conference on Research and Development in Information Retrieval (SIGIR '99)*, ACM Press, New York. 1999, pp. 230-237.

[27] H. Luo, C. Niu, R. Shen, and C. Ullrich. "A collaborative filtering framework based on both local user similarity and global user similarity," *Machine Learning*, vol.72, no.3, pp. 231-245, 2008