Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 847965, 6 pages
http://dx.doi.org/10.1155/2013/847965

Research Article

Novel Neighbor Selection Method to Improve Data Sparsity Problem in Collaborative Filtering

Hyeong-Joon Kwon and Kwang Seok Hong

College of Information and Communication Engineering, Sungkyunkwan University, 300 Chunchun-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Republic of Korea

Correspondence should be addressed to Hyeong-Joon Kwon; katsyuki@skku.edu

Received 5 March 2013; Accepted 16 July 2013

Academic Editor: Tai-hoon Kim

Memory-based collaborative filtering selects the top-k neighbors with high rank similarity in order to predict a rating for an item that the target user has not yet experienced. The most common traditional neighbor selection method for memory-based collaborative filtering is priority similarity. In this paper, we analyze various problems with the traditional neighbor selection method and propose a novel method to improve upon them. The proposed method minimizes the similarity evaluation errors with the existing neighbor selection method by considering the number of common items between two objects. The method is effective for the practical application of collaborative filtering. For validation, we analyze and compare experimental results between an existing method and the proposed method. We were able to confirm that the proposed method can improve the prediction accuracy of memory-based collaborative filtering by neighbor selection that prioritizes the number of common items.

1. Introduction

Shared online content is improving in quality and quantity due to the rapid growth of IT infrastructure over the last few years. Collaborative filtering (CF) is a technology for creating personalized recommended content lists for users from overwhelming amounts of content and is currently a widely used and successful method for recommendation system organization [1].

CF is based on the premise that a user's past content preferences will persist in the future. The most well-known method, memory-based collaborative filtering (MBCF, also known as neighbor-based CF), can be classified into user-based and item-based approaches [2, 3]. Based on a user-item rating dataset, a procedure predicts the ratings of target items not yet rated by the target user. Based on coitem ratings between users, the user-based approach calculates the similarities of all users but the target user. Then, the users are all listed in order, with those closer to the target user being the highest on the list. The prediction of the target user's target item is based on the k highest similar users' ratings. The item-based approach uses coratings of all items and target items and calculates their similarity. The item list is created based on the similarity to the target item. The highest rated k items are used to predict the rating of the target user's target item.

The most important performance evaluation criterion of MBCF is the prediction accuracy, which is estimated from the error between the CF and the actual rating. One of reasons for the decline in CF prediction accuracy is the data sparsity problem, which occurs due to an insufficient number of user ratings from the user-item rating dataset [1, 2]. This arises from an insufficient quantity of coitem ratings for similarity calculations between users or items. Various methods have been proposed to solve data sparsity problems and can be classified into various approaches. The first approach is to reduce the empty space of nonrated items by dimensionality reduction of the user-item rating dataset [4]. The second approach is to expand the framework of MBCF and to use additional information from an existing similarity method [5, 6]. The third approach is to reduce the existing similarity method's weaknesses in order to create a new similarity method [7, 8]. Another approach is to develop a novel neighbor selection method.

In this paper, we examine neighbor selection in order to reduce the data sparsity problem. The proposed method considers the number of coitems as a priority when similarity is evaluated between two objects. The proposed neighbor selection method minimizes the decrease in performance that results from a lack of the number of coitems and increases the prediction accuracy in many coitems. We have confirmed the effectiveness of the proposed method with a full-rating experiment using the MovieLens 100 K and 1 M dataset [9]. The CF application using the proposed method can be applied to a personalized information-providing system, such as those in e-commerce, video on demand, and cloud-computing-based multimedia content sharing. The proposed method improves the prediction accuracy of the CF system, which is the most important performance metric for CF.

This paper is organized as follows. In Section 2, we describe MBCF and previous studies on the traditional problem. We propose a new neighbor selection method in Section 3. In Section 4, we show the efficiency of the proposed method. In Section 5, we summarize the results of this study and introduce the application area.

2. Related Works

MBCF calculates the similarity between users and items based on the user-item rating dataset. To make a prediction for user m and item n, the user-based approach calculates the similarity of two users' coitem ratings and lists all other users in order of similarity to user m. The item-based approach then calculates the similarity between two items using their coratings and lists all other items in order of similarity [1, 2]. The vector-space-model-based method includes Cosine Similarity (COS), which is frequently used in information retrieval. COS assumes that the rating of each user is a point in a vector space and then evaluates the cosine angle between the two points. It considers the common rating vectors $X = \{x_1, x_2, x_3, \ldots, x_n\}$ and $Y = \{y_1, y_2, y_3, \ldots, y_n\}$ of users X and Y, which are represented by a dotproduct and a magnitude. COS has frequently been used for performance comparisons in CF. The cosine angle between the vectors X and Y is given by

$$\begin{aligned}\cos(X,Y) &= \frac{X \bullet Y}{\|X\| \, \|Y\|} \\ &= \frac{x_1 y_1 + x_2 y_2 \cdots x_n y_n}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}\cdots\sqrt{x_n^2 + y_n^2}}.\end{aligned} \tag{1}$$

The correlation-based method includes the Pearson dotproduct correlation coefficient (PCC) and the Spearman rank correlation coefficient (SRCC). Two correlation coefficients are normally used to evaluate the association intensity between two variables. To evaluate the correlation between items with common ratings $X = \{x_1, x_2, x_3, \ldots, x_n\}$ and $Y = \{y_1, y_2, y_3, \ldots, y_n\}$ of two users X and Y, the PCC $\gamma(X,Y)$ and SRCC $\rho(X,Y)$ are given by

$$\begin{aligned}\gamma(X,Y) &= \frac{\sum_{i=1}^{n}\left(x_i - \overline{X}\right)\left(y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \overline{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i - \overline{Y}\right)^2}}, \\ \rho(X,Y) &= \frac{6\sum_{i=1}^{n}\left(x_i - y_i\right)^2}{n\left(n^2 - 1\right)}.\end{aligned} \tag{2}$$

The PCC and SRCC are known to show consistent performance. The PCC is used more frequently than the SRCC, because the latter must determine the ranking of items in a hierarchical order with a rating score.

The raw moment similarity (RMS) was proposed in August 2011 by our research group [8]. The n value is the number of coitem ratings v between user i and user j. The variable r is the rating range in the dataset. The random variable D is composed of the absolute values d_z of the difference between the coitem ratings of a pair of users. k is a parameter of the proposed similarity measure, which is used to apply a penalty to the difference between the coitem ratings. This algorithm solves various problems of COS and PCC inaccuracy when data is sparse. The RMS is as follows:

$$\begin{aligned}\mathrm{RMS}\left(u_i, u_j\right) &= 1 - \frac{1}{r^k}\frac{1}{n}\sum_{v=1}^{n}\left(\left|u_{i,v} - u_{j,v}\right|\right)^k \\ &= 1 - \frac{1}{r^k}\sum_{z=1}^{m}\Pr\left(D = d_z\right)\cdot d_z^k.\end{aligned} \tag{3}$$

After calculating the similarity for all pairs of objects, the MBCF selects objects from a fixed number of the nearest neighbors. Accordingly, the MBCF predicts the preference rating with similarities for weights. Existing studies have calculated the mean absolute error (MAE) or root mean squared error (RMSE) between the predicted and real ratings in order to evaluate the prediction accuracy. The two measurements show the same performance graph. The MAE is preferred over RMSE for performance evaluation.

In the user-item rating matrix of MBCF, if a user m does not rate an item n, then $r_{m,n}$ is empty. High data sparsity exists when there is far more empty space than filled space in the user-item rating matrix. Empty spaces indicate few ratings of other users that can be used to predict the target user's rating. So, an insufficiency of coratings between two objects for similarity calculations results in wrong similarity or a high chance that the data sparsity problem will make similarity calculation impossible. In previous studies, the analysis result of the quantity of the dataset's pair object coratings frequently revealed cases where the quantity of coratings was insufficient. Many researchers have proposed various methods to address the problem.

The first approach is to reduce the dimension of the user-item rating matrix, while the second approach is to expand the basic framework of MBCF with additional information [5, 6]. For example, for the second approach, there are methods using additional information such as semantics or categories.

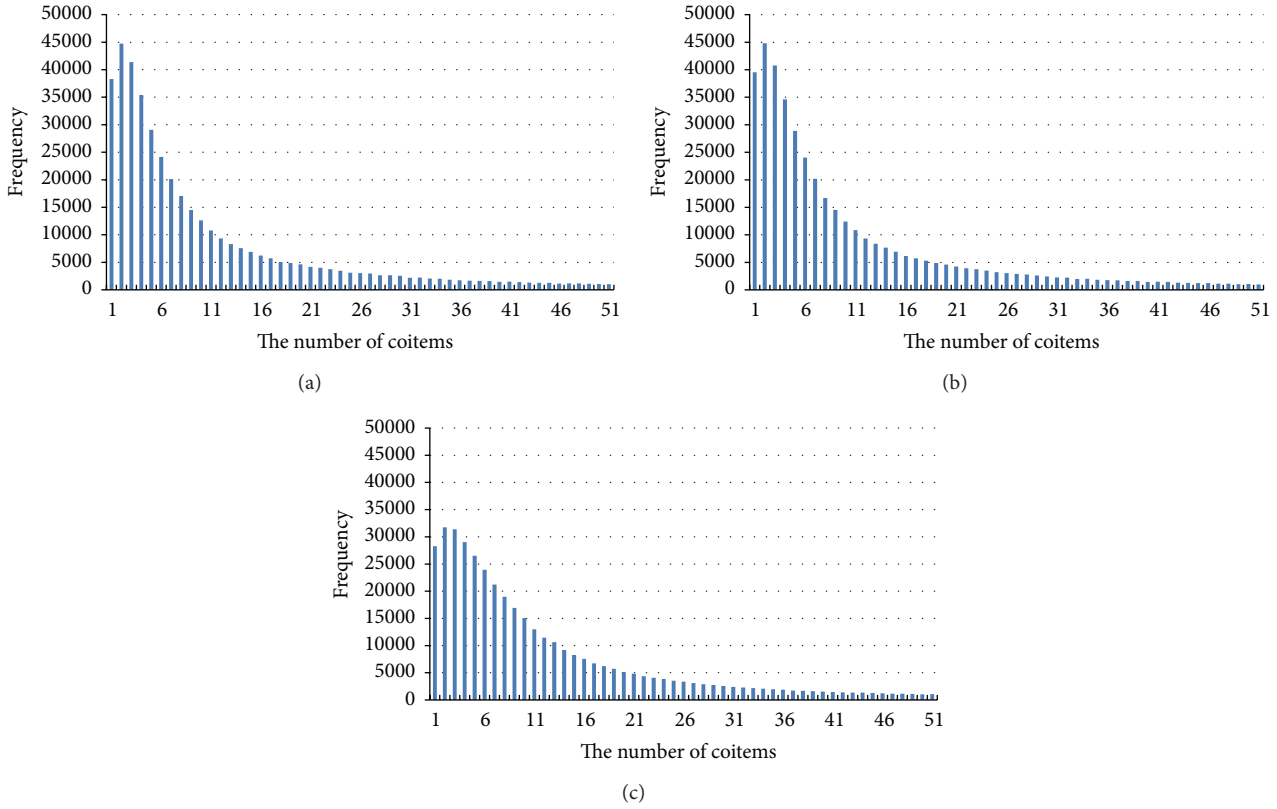

Figure 1: Coitems histogram analysis: (a) MovieLens 100 K: UA, (b) MovieLens 100 K: UB, and (c) MovieLens 100 K: UC.

The user preference difference is based on item property information or on the time flow and rating predictions of other objects similar to the target object that can be used to calculate similarity. The third approach is to develop a similarity algorithm, and the fourth approach is to develop neighbor selection methods [7, 8, 10, 11].

The major goal of these approaches is to increase the accuracy of the rating prediction by mitigating the data sparsity problem. In other studies on the data sparsity problem, analyzing the quantity of the dataset's pair object coitems frequently revealed cases when the quantity of coratings was insufficient. A recent noteworthy study presented a similarity method using proximity-impact-popularity (PIP). Ahn discussed the problems of widely used similarity methods resulting in decreased prediction accuracy and proposed PIP similarity [11]. Ahn also analyzed the cause of the cold-start and data sparsity problems for well-known similarity methods. When coratings are comparatively low, the cold-start condition can occur between two variables with linear relations. Ahn showed that frequently used similarity methods result in the wrong similarity results in the cold-start condition. The prediction performance decline of low data sparsity, such as that in the data sparsity problem in memory-based CF, is caused by the low number of coratings between users or items that show incorrect similarity prediction results. Using a similarity method that is robust to the data sparsity problem can improve upon these two problems. In addition, it would be the most radical method of problem improvement applicable to other methods for solving the problem.

3. Novel Neighbor Selection Method

The MBCF evaluates the similarity between a target object and every other object and then selects a given constant number k of the nearest neighbors. At this point, a priority of the similarity is generally considered. Previous studies asserted that PCC and COS give the wrong similarity for a few coitems, and this condition is observed sufficiently often. We drew frequency distributions from the MovieLens 100 K dataset. A histogram of similarities with the total number of pairs of users in the training set was made, and the frequency of the number of coitems was observed.

The histograms are shown in Figure 1. The x-axis of the histograms indicates the number of coitem ratings, while the y-axis indicates the frequency. The MovieLens 100 K dataset consists of 943 users and 1,682 items. Therefore, the maximum value of the x-axis is 1,682, and the maximum value of the y-axis is 444,153. The PCC and COS show wrong similarity values for a few coitems. The threshold is approximately 5 coitems or fewer. Figure 1(a) includes 411,096 items with 5 or fewer coitems, which corresponds to just 92.55% of the training results. Figure 1(b) includes 410,854 items, corresponding to just 92.50%. MovieLens 100K_UA and UB are separated by a dataset distributor. The datasets ua.base, ua.test, ub.base, and ub.test split the original data into

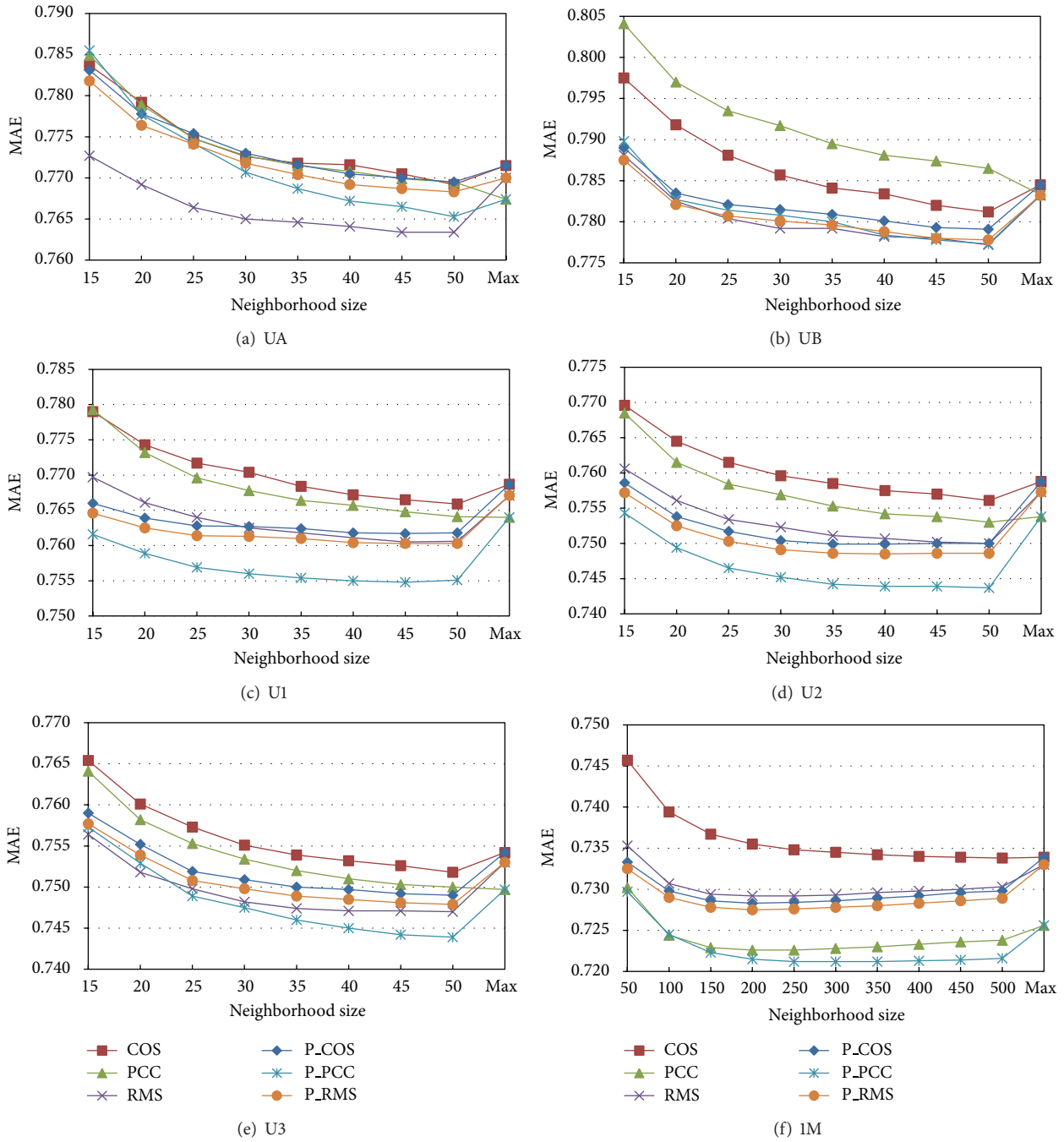

Figure 2: Experimental results using MovieLens dataset.

a training set and a test set. The test set has all items with exactly 10 ratings per user. The number of items in the test dataset is 9,430. The sets ua.test and ub.test are disjoint. For this reason, Figure 1(a) may be similar to Figure 1(b). Thus, we separated the original dataset by random sampling. A histogram of MovieLens 100K_UC generated by our research team by random sampling is shown in Figure 2. This set includes 395,673 items with 5 or fewer coitems, corresponding to just 89.08%.

The PCC gives a similarity of 0 or 1 when the number of coitems is 1 and 2. In addition, when the number of coitems is 3 and 4, it has 0 or 1 as the similarity with high probability. In other words, PCC and COS do not classify similar points between objects with few corating items. The existing neighbor selection method cannot select optimal neighbors because it considers similarity independently of the number of coitems. We have shown a critical problem of the priority of similarity via Figure 1. To address this problem, we propose a novel neighbor selection method. The proposed method first considers the number of coitems. After similarity evaluation, the traditional neighbor selection method chooses neighbors according to similarity, whereas

Table 1: A part of neighbor selection result of traditional method and proposed method using Cosine similarity.

Order	Before			After		
	Neighbor no.	Similarity	The number of coitems	Neighbor no.	Similarity	The number of coitems
1	540	0.983	18	655	0.950	165
2	874	0.980	8	234	0.938	149
3	292	0.978	57	537	0.935	148
4	473	0.975	14	417	0.940	145
5	477	0.971	5	896	0.940	137
6	457	0.968	126	201	0.939	136
7	883	0.968	97	406	0.942	130
8	886	0.966	108	387	0.944	127
9	936	0.964	43	457	0.968	126
10	334	0.960	121	758	0.957	126
11	381	0.959	43	334	0.964	121
12	184	0.959	92	533	0.947	111

Table 2: A part of neighbor selection result of traditional method and proposed method using the Pearson correlation coefficient.

Order	Before			After		
	Neighbor no.	Similarity	The number of coitems	Neighbor no.	Similarity	The number of coitems
1	381	0.575	43	655	0.195	165
2	540	0.563	18	234	0.208	149
3	886	0.493	108	537	0.209	148
4	477	0.480	5	417	0.156	145
5	566	0.462	75	896	0.343	137
6	829	0.450	25	201	0.226	136
7	334	0.440	121	406	0.278	130
8	936	0.420	43	387	0.306	127
9	483	0.419	26	457	0.392	126
10	184	0.405	92	758	0.276	126
11	457	0.392	126	334	0.440	121
12	299	0.375	106	533	0.306	111

the proposed method chooses neighbors according to the number of coitems.

Tables 1 and 2 show a part of the neighbor selection process of the proposed method and the traditional method with COS and PCC. The console print format is {target user/neighbor user/similarity between target user and neighbor user/the number of coitems between target user and neighbor user}. Because the neighbor list is different, the rating prediction result may also be different. Since the traditional method selects similar neighbors according to the CF concept, which is to "consider neighbors with similar taste," the problem of the lack of the number of coitems must be considered. The proposed method can address this problem. We next show the effectiveness of our method in an experiment. "Before" in Tables 1 and 2 indicates the neighbor selection result of the generic method, whereas "After" indicates the result of the proposed method. The traditional method sorts by similarity, while the proposed method sorts by the number of coitems.

4. Experiment and Results

We use MovieLens 100 K: UA, UB, U1, U2, and U3 and MovieLens 1 M datasets. UA and UB were explained in Section 3 and are used to verify the effectiveness of the proposed method. The MovieLens 100 K dataset consists of rating data from 100,000 items with 943 users and 1,682 items. Each user rated at least 20 items [9]. The amount of test data is 9,430, as calculated by 943∗10. We try to make 9,430 predictions and then calculate the absolute error between the real rating and the predicted rating. The sparsity level of the dataset is 93.6%, which is calculated by 1 − {100,000/(943∗1,682)}.

A full-rating experiment of MovieLens 100 K: UA and UB using the traditional neighbor selection method is shown in Figures 2(a) and 2(b). RMS shows the best performance at 0.7634. The RMS is robust to datasets with a high sparsity level [6]. In other words, it is effective even with only a few coitems. In comparison, PCC is known to be robust

when there are many coitems. PCC is accurate with objects with many coitems according to previous studies. Therefore, the proposed method should decrease the MAE of PCC, because it chooses neighbors with many coitems before those with high similarity. With this approach, the MAE of the RMS may be increased. Figures 2(a) and 2(b) show the validating experimental results of the proposed method. We can confirm that the MAE of the PCC is decreased and that the MAE of the RMS is increased. This is because RMS is already robust to having few coitems.

UA and UB are the random samplings of 10 test data from each user, while U1, U2, and U3 are randomly sampled from 30% of all data. All samplings of UA, UB, U1, U2, and U3 are from the dataset provider. We have confirmed the improvement of the performance using the proposed method from the experimental results of U1, U2, and U3. The efficiency of the proposed method is shown clearly in the MAE graph of PCC in Figure 2. As a result, using various test data for iterative prediction experiments on a single dataset, we have confirmed the improvement of CF prediction accuracy with the proposed method.

Furthermore, we executed an additional experiment using the MovieLens 1 M dataset, which has a sparsity level lower than that of the MovieLens 100 K dataset. Since test data is not classified by the dataset provider for the MovieLens 1 M dataset, 30% of the dataset was randomly sampled. This dataset contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users, who joined MovieLens in 2000. Each user rated at least 20 items. The amount of test data is 200,042, which are randomly sampled from the dataset. We try to make 200,042 predictions and then calculate the absolute error between the real rating and the predicted rating. The sparsity level of the dataset is 57.5%, which is calculated by $1 - \{10{,}000{,}209/(3{,}900 * 6{,}040)\}$. As shown in Figure 2(f), with a lower sparsity level, the PCC with low MAE has even lower performance using proposed method. To summarize, the experiment using MovieLens 100 K and 1 M has proven that the proposed method can improve the performance of prediction accuracy, regardless of its sparsity level.

5. Conclusion

We have suggested a novel neighbor selection method for MBCF. The proposed neighbor selection method improves the prediction performance of the MBCF by prioritizing the number of coitems. Our approach addresses the weakness of the existing method with a priority of similarity, which is used to show wrong similarity by PCC and COS. In future work, we will study a combination of the proposed neighbor selection method and a priority of similarity. We expect that the future work can improve upon the data sparsity problem with few coitems and contribute to the improvement of performance in many coitems. The proposed method can be applied in various areas. The application range is the same as that of CF-based recommender systems, including e-commerce, location-based systems, video on demand, smart TV, and movie clip sharing services.

Acknowledgments

This research was supported by MSIP, Republic of Korea, under ITRC NIPA-2013-(H0301-13-3001) and PRCP through NRF of Republic of Korea, funded by MOE (NRF-2010-0020210).

References

[1] 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.

[2] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 203–237, ACM Press, Berkeley, Calif, USA, 1999.

[3] B. Sawar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the International Conference on World Wide Web (WWW '01)*, pp. 285–295, ACM Press, Hong Kong, China, 2001.

[4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: a constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[5] L. Yu, L. Liu, and X. Li, "A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce," *Expert Systems with Applications*, vol. 28, no. 1, pp. 67–77, 2005.

[6] H.-J. Kwon and K.-S. Hong, "Personalized electronic program guide for IPTV based on collaborative filtering with novel similarity method," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '11)*, pp. 467–468, Las Vegas, Nev, USA, January 2011.

[7] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[8] H.-J. Kwon and K.-S. Hong, "Personalized smart TV program recommender based on collaborative filtering and a novel similarity method," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1416–1423, 2011.

[9] Movielens Dataset, GroupLens Research at the University of Minnesota, http://www.grouplens.org/node/73.

[10] T.-H. Kim and S.-B. Yang, "An improved neighbor selection algorithm in collaborative filtering," *IEICE Transactions on Information and Systems*, vol. 88, no. 5, pp. 1072–1076, 2005.

[11] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.