International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.9 (2014), pp.357-370
http://dx.doi.org/10.14257/ijmue.2014.9.9.36

Social Clustering-based Similar User Indexing for Large Recommender System

Haesung Lee[1] and Joonhee Kwon[2]

[1, 2]Department of Computer Science, Kyonggi University,
San 94-6, Yiui-dong, Yeongtong-ku, Suwon-si, Gyeonggi-do, Korea
{seastar0202, kwonjh}@kgu.ac.kr

Abstract

The tremendous growth of data in recent years poses some key challenges for recommender systems. Theses keys are related with producing high quality recommendations and fast performing the composition recommended items. In this paper, we propose social clustering-based similar user index to not only improve the prediction of recommendations, but also compose personalized recommendations in fast. Through the experimental result, we show that proposed clustering method is more accurate than k-means which is prevalent clustering techniques. And, we reduce computation time needed for composing recommendation. That is, proposed clustering-based indexing method improves the performance of recommender systems which deals with a very large data.

Keywords: *Large Recommender system, clustering, graph theory, automatically personalized item delivery services*

1. Introduction

With the exponentially increasing amount of information in the web, the recommender systems have been advanced in research and e-commerce site areas. The goal of recommender systems is to recommend items or products that fit a user's tastes, in order to help the user in selecting or purchasing items from an overwhelming set of choices [1]. Personalized recommendations are especially important in e-commerce sites where the variety of choices is large, the taste of the customer is important. Some of the major e-commerce sites, like Amazon and Netflix, successfully apply recommender systems to deliver automatically generated personalized recommendation to their customer. One of the earliest and most successful recommender technologies is collaborative filtering (CF). CF has been very successful in both research and practice. Collaborative filtering approach to recommender systems predicts user preferences for products or services by learning past user's history data [2]. In other words, most CF-based recommender algorithms are usually designed to work on various data sets such as products or rating values for products experienced by users. Amazon, for example, that has incorporated CF-based recommender systems to personalize the e-commerce site for each user, has recorded more than 30 million users and several million products [3].

Many recommender techniques have been developed in the past decade, but a considerable amount of them were constructed with small datasets and they are entirely unrealistic attempts. While the tremendous growth of these data sets in recent years poses some key challenges, several recommender systems suffer from performance and scalability problems when dealing with larger datasets. Theses keys are mainly related

ISSN: 1975-0080 IJMUE

with two challenges, producing high quality recommendations and fast performing the composition recommended items dealing with the very large data set.

With the exploding information in e-commerce site, there are needs for new technique for the large recommender systems. The large recommender systems are able to search tens of millions of items in real-time. However, existing CF-based algorithms have performance problems with individual users for whom the site has large amount of history data. So, the first challenge is to compose personalized recommendations in fast way. Users should not use slow recommender systems. The second challenge is that the large recommender systems have to propose each user accurate recommendation because users need recommendations they can trust to help them find products they will like. In some ways these two challenges are in conflict, since the less time the recommender system spends composing recommendations, the worse the quality of recommendations. For this reason, it is important to treat the two challenges simultaneously. So the solutions for both challenges have to be useful and practical.

In this paper, we propose social clustering-based similar user index. With the use of the similar user index, we efficiently reduce computation time for composing recommendations. And, we define newly social clustering method in order to improve the accuracy of recommendations. Generated clusters are considered as the similar user index. Also, in the proposed social clustering, we apply the concept of the affiliation network among the graph theory. Based on these graph theory, it is possible to generate a social network from user's history data. Besides, we improve the accuracy of clustering similar users.

The rest of this paper is organized as follows. Section 2 describes related works and background of proposed methods for generating the social clustering-based similar user index. Section 3 discusses in detail the social clustering, the indexing mechanism and the implementation of social clustering-based similar user index. Section 4 presents the experimental study and the results. Finally, Section 5 concludes this paper and gives an outlook upon our future research in this area.

2. Related Works and Backgrounds

2.1. Collaborative Filtering

Collaborative filtering (CF) is the most successful recommendation technology and is used in many of the most successful recommender systems. CF systems employ statistical techniques to find a set of similar users who either experience different items similarly or tend to buy similar set of items [2]. Once a set of similar users is formed, these systems calculate prediction expressing the predicted preference score of item i_j based on formed similar users. Then, CF systems produce recommendations which are a list of *N* items that the active user will like the most. The recommended list usually consists of items not already experienced by the active user. CF algorithms have been successful in several domains, but the algorithm is reported have shown some limitations such as sparsity and scalability [2].

For composing a set of similar users, CF algorithms rely upon exact matches between user's past experienced item history. Therefore, if there is sparse rating data for matching between users, the accuracy of user's predicted preference is reduced. Ultimately, the performance of recommender systems is lower. And, composing the set of similar users requires computation that grows with both the number of users and the number of items. With millions of users and items, most CF-based recommender systems suffer serious scalability problems. So, sparsity and scalability are the main limitations of CF algorithms.

2.2. Clustering

Clustering is a kind of data mining technique for discovering interesting patterns from a given database [4]. The main idea of clustering is that given n data pointing in a m-dimensional metric space is divided into k clusters so that all the data pointing within one cluster has a closer similarity than the data within any other cluster. There are various clustering methods and they are currently widely used. Among them, *K*-means method is a widely used clustering procedure that searches for nearly optimal partition with a fixed number of clusters [5]. The *K*-means algorithms have been popular because of its easiness and simplicity for application. In the domain of recommender systems, *K*-means clustering techniques work by identifying groups of users who appear to have similar preferences. Once the clustering is complete, the performance of recommender systems can be very good, since the size of data that must be analyzed much smaller. Although, the clustering result may depend on the initial seeds, however, there are few studies for the mechanism which optimize the initial seeds. Ultimately, clustering techniques usually produce less-personal recommendation than other methods and most often lead to worse accuracy than CF algorithms.

2.3. Recommender Algorithms with the Social Network

[6] presented a novel framework for studying recommendation algorithms in terms of the 'jumps' that they make to connect people to items. This approach emphasizes reachability via an algorithm within the implicit graph structure underlying a recommender dataset. The study approaches recommendation from a different but complementary perspective of considering the connections that are made. According to the study, most recommender systems miss many desirable aspects of the recommendation process. They assert that recommendation is an indirect way of bringing people together. In many situations, users would like to request recommendations purely based on various constraints such as social relationships on the nature of specific connections explored. That is, CF algorithms exploit connection between users and items.

Social network theory can be used to model such a recommendation system of people versus items as an affiliation network and distinguishes between a primary mode and a secondary mode, where a mode refers to a distinct set of entities that have similar attributes [7]. In the view of recommender systems, the primary mode could be regarded as users. And, the secondary mode is considered as items. In other words, in a recommender system, the rating patterns of people on items induce an implicit social network and influence connectivities in the network.

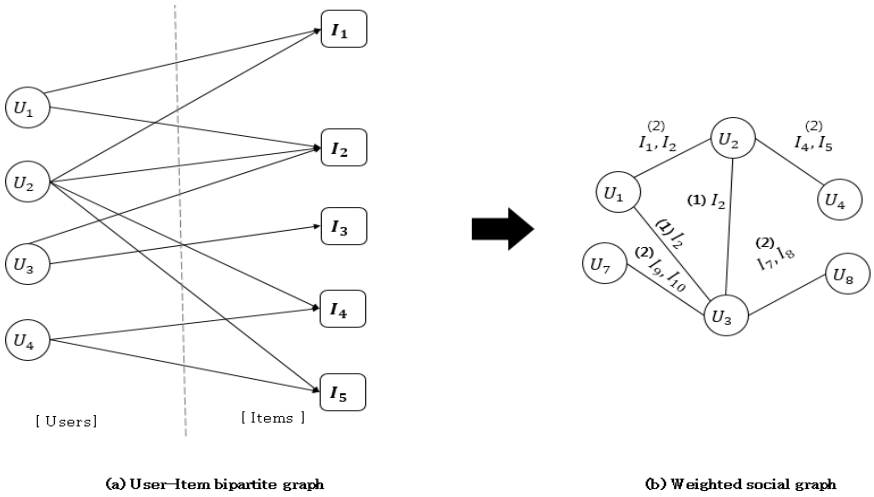
3. Social Clustering-based Similar User Indexing Mechanism

3.1. Generating Social Network with the Recommender Dataset

Based on the argument of [6], we generate a social network with the recommender dataset. With the generated social network, we compose the similar user index. The similar user index is used for improving the performance of recommender systems.

The recommender dataset is composed of user's profile, item's metadata and the rating for items. We understand the fact that there are relationships between a user and an item on which the user assign the rating value with their preference. According to the social network theory [8], we consider the user as the primary mode and the item as the secondary mode. Based on this concept, we deduct social network model from the recommender dataset. A recommender dataset $\mathcal{R}$can be represented as a bipartite graph $G = (U, I, E)$like shown Figure

1(a), where *U* is the set of people, *I* is the set of items, and the edges in *E* represent the ratings of items. With the concept of the affiliation network in the graph theory, we also make the assumption that social relationships can be composed if node *B* can be reached from *A* in one edge, and *C* can be reached from *B* in one edge, then *C* is reachable from *A* in two edges. Then, we consider the edge is the skip, which connects two members in *U* if they experience one item in common. The skip induces a social network graph. The social network graph of a recommender dataset $\mathcal{R}$ induced by a given skip $\mathcal{L}$ is an undirected and weighted graph$G_S = (U, E_S)$like shown Figure 1(b), where the edges are given by $E_S = \mathcal{L}(\mathcal{R})$. Ultimately, Figure 1(b) shows the social network graph induced from the example in Figure 1(a).



(a) User–Item bipartite graph (b) Weighted social graph

Figure 1. The Example of the Social Network Generated from Recommender Dataset

For example, in Figure 1(a), two users U_1 and U_2 experience itemsI_1 and I_2 in common. So, the social relationship is generated betweenU_1andU_2 as shown in Figure 2(b).

3.2. Three Steps-clustering for Composing the Similar User Index

There are various clustering techniques such as *K*-means or Min-hash [9]. Especially, *K*-means is widely used because it's easiness and usefulness. However, existing clustering techniques have fatal limitation which usually produces less-personal recommendations than other methods and most often lead to worse accuracy. Once the clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller.

For constructing the similar user index, we propose the new clustering technique based on *K*-means. The proposed clustering technique assigns a pair of users to the same cluster with probability proportional to the similarity between these users. The use of the similar index makes it much more efficient to perform the composing recommendation more quickly reducing the difficulty of the computation which is needed for making up recommendation.

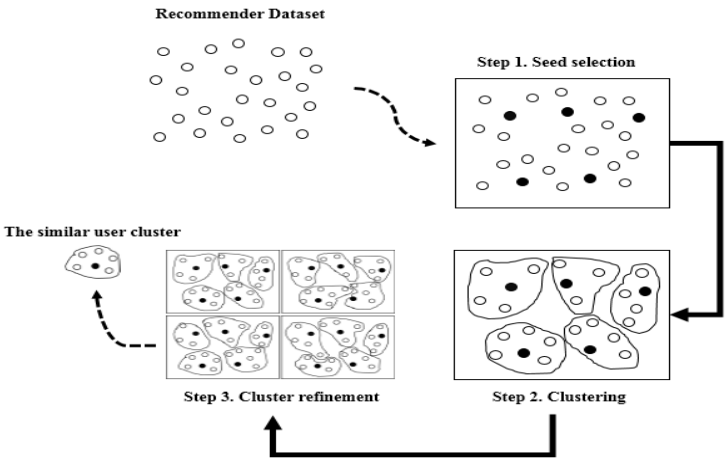


Figure 2. Three Phases of Similar User Clustering

For improving the accuracy of clustering which assigns similar user in the same cluster, we take three phases in the clustering. Figure 2 shows three phases of similar user clustering simply.

3.2.1. Initial Seed Selection

For the constructing the user similar index, we use clustering technique. The algorithm of choice for many clustering tasks is the *K*-means algorithm. The *K*-means algorithm attempts to find the cluster centers, such that the sum of squared distances of each data point to its nearest cluster center is minimized. Unfortunately, the *K*-means algorithm is a local optimization strategy and as such is sensitive to the choice of the initial position of the cluster [10]. These initial center locations are often termed the seeds for the *K*-means algorithm. General *K*-means algorithms randomly select initial seeds, which is a main cause of decreasing the accuracy of the clustering. Therefore, there are many studies about the selection of initial seeds to increase the accuracy of clustering. In the domain of recommender systems, however, there are few studies which consider the feature of the recommender systems. In this paper, we propose new seeds selection mechanism for improving the accuracy of clustering similar users.

In order to choice seeds of each cluster, we use the social network which is depicted in above the section 3.1. In the social network, the weight of relationship between users means the frequency of common items which are experienced by both users. Based on the concept of general CF, the more frequency of common items experienced by pair of users, the more similar these users are. Therefore, we assume that influential users node in the social networks have much more relationships with other users. In other words, the influential user node could be considered as a seed of each similar user cluster because the influential user node have higher possibility to take relationships with any other user nodes.(1) is the equation for measuring the importance of the user node u. If a user node has higher value of $Importance_u$than other user node, the user node is considered as a seed in a cluster.

$$Importance_u = \sum_{i \in E(u)}^{N} w_i \quad (1)$$

In equation (1), *N* is the total number of edges from the user node *u*. *E*(*u*) is all edges from the user node *u* to other user nodes. w_i is the weight of edges from the user node *u*.

Algorithm 1 selects seeds which are considered as important users in the generated social network.

Algorithm 1. *Initial Seed Selection*

Input: *G* is the social network, and *k* is the number of seeds
Output: seeds
1.ω has arbitrary value which is higher than *k* or same with *k*.
2:*candidateUserSeeds*[ω]←*CentralUserNodes(G)*;
3. **if**ω ==*k***then**
4. seeds ←*candidateUserSeeds*[ω]
5: **return**seeds
6*:* **else**
7: seeds ←*SeedFiltering*(candidateUserSeeds[ω])
8: **return** seeds

In Algorithm 1, ω has arbitrary value which is higher than *k* or same with *k*. The function, *CentralUserNodes*returns the set of user nodes which have higher importance than other user nodes have. Since there is possibility to be user nodes which have same high importance values, the number of user nodes returned by *CentralUserNodes*is more than given number of seeds or same with the number of seeds. If the number of candidate seeds returned by *CentralUserNode* is higher than *k*, the function, *SeedFiltering*filters out candidate similar user nodes and only compose of *k* user nodes as seeds. *SeedFiltering* takes 2 phases of filtering.

1. If some user nodes of candidate seeds have connection with each other, the user node which has higher importance than other nodes is chosen as the seed.
2. Among seeds composed by phase 1, *k* seeds are randomly chosen being initial seeds.

While the second phase of *SeedFiltering*little impacts the accuracy of clustering, we take clustering refinement in order to improve the accuracy of clustering after the phase of clustering.

3.2.2. Similar User Clustering

Similar user clustering techniques work by identifying groups of users who appear to have similar preferences. Each seed which is selected in the phase 1 is located in the center of the cluster. Major task of similar user clustering is measuring the distance between a seed and a user node. If the distance between the user node *A* and the seed of some cluster *R* is shorter than seeds of other clusters, the user node A is included in the cluster *R*. The distance is considered as similarity of the user node and the seed. For measuring the similarity, we use the square of the Euclidean distance measurement. Equation 2 is for measuring similarity between seeds and user nodes based on Euclidean distance measurement.

$$Sim_{Seed_i,u} = \frac{1}{\sqrt{\{TN(Seed_i) - MN(Seed_i,u)\} + \{TN(u) - MN(Seed_i,u)\}}} \quad (2)$$

In equation (2), *TN* is total number of items experienced by a user node or a seed. *MN* is the number of items experienced by a pair of nodes in common. Algorithm 2 takes similar user clustering based on selected seeds.

Algorithm 2. *Similar User Clustering*

Input: *Seeds*is selected seeds, *k* is the number of clusters, *U* is the user dataset
Output: The set ofclusters , *C*
1.Assign each seed $s_k$in each cluster$c_k \in C$
2. $centroid_k \leftarrow$*Position*(s_k)
3. **repeat**
4.**for**each user node $u_i \in U$except user nodes which are seeds **do**
5. **for** each seed $s_k \in Seeds$**do**
6. $similarity_{i,k} \leftarrow Euclidean(u_i, centroid_k)$
7. $centroid_k \leftarrow ReadjustCentroid(c_k)$
8. **if** $similarity_{i,k}$ is biggest value among other clusters **then**
9. $c_k \leftarrow u_i$
10.**until** centroids of clusters do not change
11.**return***C*

In algorithm 2, $centroid_k$ is center position of cluster *k*. The function, *Position* returns the location in the cluster. We define initial $centroid_k$as the position of each seed in clusters. *Euclidean* measures the distance between the user nodes and the center position of each cluster. *Read just Centroid* read just the center position in each cluster based on recently computed similarity. Among clusters, the cluster which has the shortest distances between the user nodes u_i and the center position is chosen to include the user node u_i . The clustering is iterated until the center position of each cluster doesn't changed.

3.2.3. Clustering Refinement

In order to improve the quality and the accuracy of the similar user clustering, we take clustering refinement technique at the final step of the similar user clustering. For this, we apply the modularity technique. In the domain of information clustering, the modularity is a criterion for evaluating the quality of partitioning a network into clusters [11]. While there are various modularity technique, *Q* is widely known as the most accurate [12].

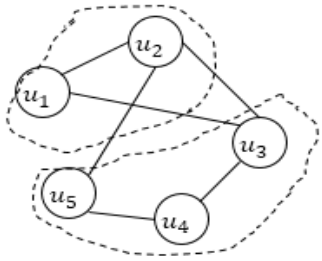
With the concept of *Q*, we suppose that particular divisions of the social network which is generated through the step 1 are considered as *k* clusters. And these divisions can be represented by a$k \times k$ symmetric matrix *e*in which each element $e_{ij}$is the fraction of all edges that links user nodes in cluster $c_i$to clusterc_j. Equation (3) is for measuring *Q*of the similar user clusters. If the result of clustering has the high value of *Q*, this is considered that relatively optimized and accurate clusters are constructed.

$$\mathrm{Q} = \sum(e_{ij} - a_i^2) = Tr(e) - \|e^2\| \tag{3}$$

In equation (3), *Tr*(*e*) represents a fraction of edges that connect the user nodes in a cluster and obviously a good cluster or division has a high value of *Tr*(*e*). a_i^2 is the expected fraction of edges within the cluster $c_i$when the edges were distributed randomly on the social network. $\|e\|$ is the sum of matrix *e* elements.

Figure 2 shows a small example. In Figure 2(a), there is a social network which has five user nodes and two clusters $C_1 = \{u_1, u_2\}$and $C_2 = \{u_3, u_4, u_5\}$.Fig. 2(b) represents

the matrix which is transformed from the social network shown in Figure 2(a). The $u_i$represents each user. The e_{ij}values are the sums of matrix elements belonging to a pair of $C_i$and C_j divided by total sum of all matrix elements. For example, in Figure 2, values of e_{ij} are calculated as e_{11} = 2/12, e_{12} =2/12, e_{21} =2/12 and e_{22} =4/12. The modularity of clustering the example social network into two cluster is $Q = (e_{11} - a_1^2) + (e_{22} - a_2^2)$ =((2/12)-((5/12)^2)+ (((4/12) – (7/12)^2)=-(1/72). The negative value of Q clearly shows a suboptimal partition.



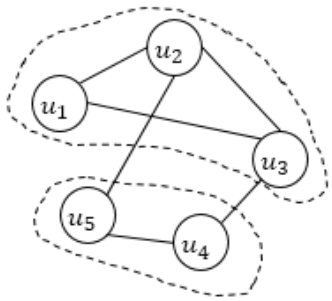
(a)

		C_1		C_2		
		u_2	u_3	u_1	u_4	u_5
C_1	u_2	0	1	1	0	1
	u_3	1	0	1	1	1
C_2	u_1	1	1	0	0	0
	u_4	0	1	0	0	1
	u_5	1	0	0	1	0

(b)

Figure 2. The Example of Social Network Clustering [11]

For improving$Q$, the user node u_3 included in the cluster $C_2$is arbitrarily selected, and moved from the cluster C_2 to the cluster C_1like shown Figure 3(a). Then, values of e_{ij} are calculated again: e_{11}= 6/12, e_{12}=2/12, e_{21}=2/12 and e_{22}=2/12. Figure 3(b) shows a matrix representing the social network of Figure 3(a).



		C_1			C_2	
		u_1	u_2	u_3	u_4	u_5
C_1	u_1	0	1	1	0	0
	u_2	1	0	1	0	1
	u_3	1	1	0	1	0
C_2	u_4	0	0	1	0	1
	u_5	0	1	0	1	0

Figure 3. The Example of Clustering Refinement

Based on the matrix of Figure 3(b), The modularity of clustering the example social network is $Q = (e_{11} - a_1^2) + (e_{22} - a_2^2)$ =((6/12)-((5/12)^2)+ (((2/12) – (7/12)^2)=(1/9). That is, assigning the user node u_3 to C_1 improves$Q$from -1/72 to 1/9. So, we take clustering refinement based on Q in order to compose more accuracy similar user clusters.

Algorithm 3 returns the refined set of clusters. In order words, through the step of clustering refinement, we could compose more accurate clusters than them made in the previous step 2.

Algorithm 3. *Clustering refinement*

Input: The set of clusters, *C*, the number of clustering refinement, θ
Output: The set ofclusters , *C*
1.*iteration*← 0
2. *previousQ* ←*Q*(*C*)
3. **while***iteration*>θ
4. *iteration*++
5. *updatedC*←*ClusterRefinement*(*C*)
6. *updatedQ*←*Q*(*C*)
7. **if** *previousQ*<*updatedQ***then**
8. *previousQ*←*updatedQ*
9. *C*← *updatedC*
10.**end if**
11. **end while**
12. **return***C*

In Algorithm 3, the function, *ClusterRefinement* randomly selects two user nodes included in different clusters and exchange their cluster ID. Eventually, *ClusterRefinement* returns a readjusted matrix which represents the clusters as like shown Figure 3(b). The function, *Q* measures the modularity of the set of clusters. During the given number of iterationsθ, both functions, *ClusterRefinement* and *Q* are called repeatedly.

3.3. Searching Similar Users with the Similar User Index

In our previous study [13], we verified that computing similarity between a user and other users is the most time-consumed factor among other operation of CF-based recommender systems. To improve the performance of large recommender system, we use similar user index generated through three steps of clustering mechanism presented above in this paper.

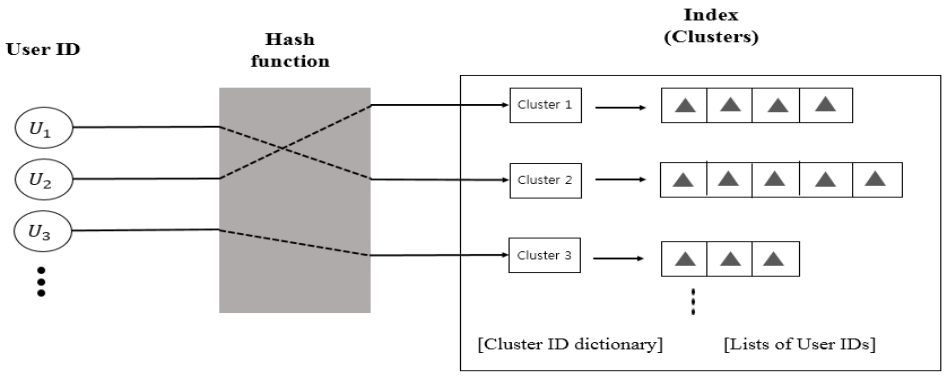


Figure 4. The Access to the Similar User Index

With the use of similar user index, the time of looking for similar users is efficiently reduced. Since each similar user cluster includes similar users, the recommender algorithms only takes data of users who are belonged in the targeted cluster without considering data about other users who are included in different clusters.

Similar clusters have their own cluster ID. And then, we use these cluster IDs as identifiers of index. For fast access to similar user index, we define a hash function. Fig. 4 depicts when a user ID given, similar users of this user ID are retrieved through the hash function. Retrieved set of similar users means a similar user cluster. Similar user index consist of a cluster ID dictionary and lists of user IDs. The dictionary of cluster IDs records cluster IDs with their own value of the centroid. Defined hash function computes similarities between the given user ID and the centroid of each similar user cluster and returns a cluster ID with which the given user has the greatest value of the similarity.

4. Evaluation

We evaluate the effectiveness of our similar user clustering-based indexing mechanism for large recommender systems in two parts. First, we show the experimental result about the response time of the recommender system in which the similar user index is used. Second, we verify the accuracy of similar user clustering through the comparative experiment. All developed prototype systems in the evaluation are implemented with Intel® Core™ CPU at 3.4 GHz, and 8 GB RAM. And we implemented prototypes with Window 7 Enterprise K, JDK 7, and apache mahout. Apache mahout is machine learning library and proposes useful libraries for recommender systems [14].

Table 1. Artificially Generated Data Sets

Name	The number of user IDs	Size(MB)
D1	3,706	1.4
D2	3,580	0.97
D3	3,430	0.85
D4	3,310	0.78
D5	2,940	0.71
D6	2,610	0.67
D7	2,470	0.58
D8	2,310	0.51
D9	2,005	0.48
D10	1,850	0.41

For the evaluation, we use the data set of MovieLens [14]. The total size of data set is about 1Mega byte. Based on the dataset, we artificially generate 10 data set which are vary with the data size. Table 1 shows 10 generated data sets.

4.1. Response Time

In this paper, we define that the response time of recommender systems is the needed time for composing recommendations for a specific user. In other words, if a recommender system has short response time, the performance of the recommender system is good in terms of speed for composing recommendations. For evaluating the performance about the response

time, we implemented two recommender systems. The one, *CF* is implemented with the general collaborative filtering algorithm and does not use the similar user index. And the other one, *proposed* uses our proposed similar user index. Figure 5 shows the comparative experimental results of *CF* and *proposed*.

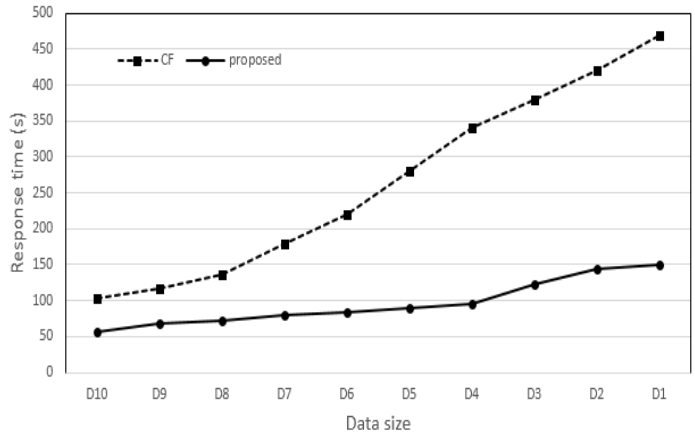


Figure 5. The Experimental Result about the Response Times with Various Data Sizes

With the experimental result like shown in Figure 5, we know that using the similar user index helps recommender systems more quickly compose recommendations than other systems which do not use the similar index. Besides, the system which uses the similar user index is relatively not affected by the data size in composing recommendations.

4.2. Clustering Accuracy

In order to evaluate the accuracy of proposed similar user clustering mechanism, we perform the comparative experiment of two clustering algorithms. Two clustering algorithms are the proposed technique and *K*-means. *K*-means is the most widely known clustering technique because of its easiness and effectiveness.

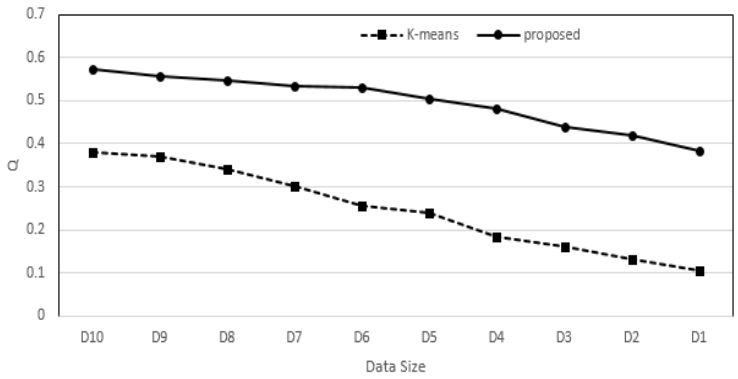


Figure 6. The Experimental Result about the Clustering Accuracy

As described in Section 3.2, proposed similar user clustering technique performs three-step clustering; 1) initial seed selection 2) clustering 3) clustering refinement. On the other hand, *K*-means randomly selects *k* seeds and takes clustering adjustment.

Both two algorithms perform clustering similar users who are consisted of the social network generated with our proposed method presented in Section 3.1. Also, both algorithms are given 10 frequency of clustering iterations. For the verification of clustering accuracy, we measure *Q* of each clustering algorithm. Figure 6 shows the experimental result of clustering accuracy. Through the experiment about the clustering accuracy, we known that our proposed method is more accurate than *k*-means in clustering similar users.

5. Conclusion

In this paper, we propose social clustering-based similar user index. With the use of the similar user index, we efficiently reduce computation time for composing personalized recommendation. Also, we define newly social clustering method with which the similar user clusters are created in order to improve the accuracy of recommendations. For the social clustering, we consider the concept of the affiliation network and construct a social network from the recommender data set. Based on the generated social network, the similar user clustering is performed. To improve the performance of large recommender system, we use similar user index generated through three steps of clustering mechanism. Since each similar user cluster includes similar users, the recommender algorithms only takes data of users who are belonged in the targeted cluster without considering data about other users who are included in different clusters.

With the result of the experiment, we know that our proposed similar user clustering mechanism is more accurate than existing clustering algorithms. Besides, the use of similar user index which are composed of similar user clusters help the large recommender system efficiently and quickly composes personalized recommendations.

Acknowledgement

This work was supported by the Gyonggi Regional Research Center (GRRC) of Korea and Contents Convergence Software (CCS) research center of Korea.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutierrez. Recommender systems survey. Konwledge-Based Systems, 46(0), pp. 109-132(2013)
[2] J. S. Breese, D. Heckerman and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering”, In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, (**1998**), pp. 43-52.
[3] Amazon, www.amazon.com
[4] R. C. Dubes and A. K. Jain, “Algorithms for Clustering Data”, Prentice-Hall, Englewood Cliffs, (**1988**).
[5] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A k-means clustering algorithm”, Applied statistics, (**1979**), pp. 100-108.
[6] B. J. Mirza, B. J. Keller and N. Ramakrishnan, “Studying recommendation algorithms by graph analysis”, Journal of Intelligent Information Systems, vol. 20, no. 2, (**2003**), pp. 131-160.
[7] M. F. Schwartz and D. C. M. Wood, “Discovering Shared Interests Using Graph Analysis”, Communications of the ACM, vol. 36, no. 8, (**1993**), pp.78-89.
[8] D. Robalino and M. Gibney, “A model of the Impact on Movie Demand of Social Networks and Word of Mouth Recommendation”, URL:http://www.metaculture.net/Searches/movies, (**1999**).
[9] S. Guha, R. Rastogi and K. Shim, “CURE: an efficient clustering algorithm for large databases”, ACM SIGMOD Record, vol. 27, no. 2, ACM, (**1998**).

[10] R. Ostrovsky, "The effectiveness of Lloyd-type methods for the k-means problem", Foundations of Computer Science, 2006, FOCS'06, 47th Annual IEEE Symposium on. IEEE, (**2006**).
[11] M. EJ Newman and M. Girvan, "Finding and evaluating community structure in networks", Physical review E vol. 69, no. 2, (**2004**).
[12] Xu, Xiaowei, N. Yuruk, Z. Feng and T. AJ Schweiger, "SCAN: a structural clustering algorithm for networks", In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, (**2007**), pp. 824-833, ACM.
[13] H. Lee and J. Kwon, "Efficient Recommender System based on Graph Data for Multimedia Application", International Journal of Multimedia & Ubiquitous Engineering, vol. 8, no. 4, (**2013**).
[14] Apache mahout, https://mahout.apache.org/
[15] MovieLens datasets, URL:http://grouplens.org/datasets/movielens/

Authors

Hae-Sung Lee, she is a Ph.D. candidate of Computer Science at Kyonggi University, Korea. Her research areas include Context-aware Computing, Social Network, Information Retrieval and Mobile Computing. She works for software development on areas of data search in ubiquitous environment. She received her B.S., M.S. in Computer Science from Kyonggi University, Korea. Contact her at seastar0202@ kyonggi.ac.kr.

Joon-Hee Kwon, she is an associate professor of Computer Science at Kyonggi University, Korea. She was a visiting research professor at the Computer Science Department at New Jersey Institute of Technology. Her research areas include Context-aware Computing, Information Retrieval, Social Network, Web 2.0 and Mobile Database. Her research projects focus on areas of d1ata search using social network and Web 2.0 in ubiquitous environment. She received her B.S., M.S. and Ph.D. in Computer Science from Sookmyung Women's University, Korea. Contact her at kwonjh@kyonggi.ac.kr.