클러스터링기반 협동적필터링을 위한 정제된 이웃 선정 알고리즘

김 택 헌[†] · 양 성 봉[††]

요 약

전자상거래에서 취급되는 상품은 오프라인 상에서 뿐만 아니라 온라인 상에서도 그 종류가 매우 다양하고 수 또한 셀 수 없을 정도로 많다. 이런 이유로 고객들이 그들의 요구에 따른 가장 적합한 상품을 찾기란 쉬운 일이 아니다. 따라서 다양한 성향을 갖는 고객들에게 더 좋은 가치를 갖는 양질의 정보를 제공하기 위해서는 고객들의 선호도를 정확하게 예측하는 능력을 갖는 개인화된 추천 시스템의 개발이 필요하다. 본 논문에서는 추천 시스템에서 클러스터링을 기반으로 한 협동적 필터링을 위한 정제된 이웃선정 방법을 제안한다. 이 방법은 그래프 접근법을 이용하며, 고객에게 영향을 줄 수 있는 다른 고객들의 집합을 보다 효율적으로 찾아낸다. 제안한 방법은 또한 서열화된 클러스터링 및 유사 가중치를 이용하여 탐색을 수행하여 보다 유용한 이웃을 선정한다. 실험 결과는 본 논문에서 제안한 방법을 이용한 추천 시스템이 보다 유용한 이웃 고객들을 찾아냄으로써 추천 시스템의 예측의 질을 향상시켜 주는 것을 보여준다.

키워드 : 추천 시스템, 협동적 필터링, 이웃 선정 방법, 클러스터링

A Refined Neighbor Selection Algorithm for Clustering-Based Collaborative Filtering

Taek-Hun Kim[†] · Sung-Bong Yang[††]

ABSTRACT

It is not easy for the customers to search the valuable information on the goods among countless items available in the Internet. In order to save time and efforts in searching the goods the customers want, it is very important for a recommender system to have a capability to predict accurately customers' preferences. In this paper we present a refined neighbor selection algorithm for clustering based collaborative filtering in recommender systems. The algorithm exploits a graph approach and searches more efficiently for set of influential customers with respect to a given customer; it searches with concepts of weighted similarity and ranked clustering. The experimental results show that the recommender systems using the proposed method find the proper neighbors and give a good prediction quality.

Key Words : Recommender systems, Collaborative Filtering, Neighbor Selection Method, Clustering

1. Introduction

There are a myriad of goods traded in commercial transactions both online and off line. These goods include physical products as well as digital contents. It is not easy for a customer to search the best suitable goods among almost countless items in the Internet. In order to provide valuable information on the items to the customers who have various preferences, we need an effective personalized recommender system.

A personalized recommender system predicts the best suitable goods for the customers according to their individual preferences and recommends the predicted results to them. Thus the customers can save time and efforts in searching the items they want.

It is very important for a recommender system to have a capability to predict accurately by analyzing the preferences of the customers. A recommender system utilizes in general an information filtering technique called collaborative filtering, which is based on the ratings of other customers who have similar preferences and is widely used for many online commercial web sites[1, 3, 5, 9, 13, 14].

A recommender system using collaborative filtering which we call it CF, calculates the similarity between the test customer who is supposed to obtain a recommendation

† 정 회 원 : 연세대학교 컴퓨터과학과 BK21 연구교수
†† 종신회원 : 연세대학교 컴퓨터과학과 교수
논문접수 : 2007년 3월 15일, 심사완료 : 2007년 6월 13일

from the recommendation system and each of other customers who have rated the items that are already rated by the test customer. Since CF is based on the ratings of the neighbors who have similar preferences, it is very important to select the neighbors properly to improve prediction quality.

There have been many investigations in selecting proper neighbors based on neighbor selection methods such as the k-nearest neighbor selection, the threshold based neighbor selection, and the clustering based neighbor selection. They are quite popular techniques for recommender systems. These techniques then predict customer's preferences for the items based on the results of the neighbors' evaluation on the same items[1, 2, 3, 10, 12].

With millions of customers and items, a recommender system running on an existing method will suffer seriously the scalability problem. Therefore there are demands for a new approach that can quickly produce high quality predictions and also can resolve the very large scale problem. Clustering based method often leads to worse prediction accuracy than other methods. Once clustering is done, performance can be quite good, since the size of a cluster that must be analyzed is much smaller[2, 11]. Therefore, the clustering based CF can be a choice to solve the very large scale problem in recommender systems.

In this paper we show that the recommender systems using a refined neighbor selection based on a graph approach. The algorithm searches with concepts of weighted similarity and ranked clustering. The graph approach allows us to exploit the transitivity of similarities. To improve prediction quality we consider both high and low similarities of each customer because customers who have contradicting similarities may give valuable information in prediction[4].

The proposed algorithm applies the clustering method to the input dataset for finding a handful of promising clusters each of which holds a set of customers with influential similarities for the test customer. It then searches the promising clusters, as if they are graph, in a breadth-first manner to extract certain customers, called *the neighbors*, who have either higher similarities or lower similarities with respect to the test customer. A cluster is considered as a graph in which vertices are customers and each edge has a weight representing the similarity between the end points(customers) of the edge.

During the search for the neighbors, we adjust the weight of each edge in the graphs according to the search depth so that the closer customers in terms of similarities have better chances to become neighbors of the test customers. The algorithm ranks the clusters according to the similarity between the test customer and the representative of each cluster after clustering. For selecting the meaningful neighbors the algorithm select the best top n clusters and merge them for the search.

In the experiments the EachMovie dataset of the Compaq Computer Corporation has been used[6]. The dataset consists of 2,811,983 preferences for 1,628 movies rated by 72,916 customers explicitly.

The experimental results show that the proposed recommender system selects meaningful neighbors for the high prediction quality. Therefore the clustering-based recommender systems using the proposed algorithms could resolve the very large scale dataset problem with high prediction quality.

The rest of this paper is organized as follows. Section 2 describes CF based recommender systems and clustering-based CF. In Section 3, the proposed refined neighbor selection algorithm is presented and in Section 4, the experimental results are illustrated. Finally, the conclusions are given in Section 5.

2. CF-based Recommender Systems

2.1 Recommender Systems

Recommender systems have been recognized as one of the solutions to reduce information overload in the Internet environment. Therefore, in online commercial transactions, if a recommender system has a capability of providing the information on the best suitable goods for the customers, they could have great satisfaction on the transactions.

A recommender system is in general based on CF and it also uses other techniques like clustering together to improve the prediction quality and the system performance. These techniques for a recommender system enhance the satisfaction of all participants such as the buyers and the sellers, so that the commercial market can be more activated.

2.2 Collaborative Filtering

A CF provided by the GroupLens plays a crucial role in reducing the customer's burden of searching[9, 14]. It filters necessary information and provides it to the customers. CF compares customers based on their previous preferences on the items to make recommendations to similar customers. So it is widely used in the recommender systems and is also called '*social*' information filtering.

CF recommends items through building the profiles of the customers from their preferences for each item. In CF, preferences are represented generally as numeric values which are rated by customers. Predicting a preference

for a certain item that is new to the test customer is based on the ratings of other customers for the 'target' item. Therefore, it is very important to find a set of customers with more similar preferences to the test customer for better prediction quality.

In CF, Equation (1) is used to predict the preference of a customer. Note that in the following equation $w_{a,k}$ is the Pearson correlation coefficient[3, 7, 9, 10, 11, 14].

$$p_{a,i} = \bar{r_a} + \frac{\sum_k \{w_{a,k} \times (r_{k,i} - \bar{r_k})\}}{\sum_k |w_{a,k}|},$$

$$\text{where } w_{a,k} = \frac{\sum_j (r_{a,j} - \bar{r_a})(r_{k,j} - \bar{r_k})}{\sqrt{\sum_j (r_{a,j} - \bar{r_a})^2 \sum_j (r_{k,j} - \bar{r_k})^2}}. \quad (1)$$

In the above equation, $P_{a,i}$ is the preference of customer a with respect to item i. $\bar{r_a}$ and $\bar{r_k}$ are the averages of customer a's ratings and customer k's ratings, respectively. $r_{k,i}$ and $r_{k,j}$ are customer k's ratings for items i and j, respectively, and $r_{a,j}$ is customer a's rating for item j.

If customers a and k have similar ratings for an item, then $w_{a,k} > 0$. $|w_{a,k}|$ indicates how much customer a tends to agree with customer k on the items that both customers have already rated. If they have opposite ratings for an item, then $w_{a,k} < 0$ and $|w_{a,k}|$ indicates how much they tend to disagree on the item that both again have already rated. Hence, if they don't correlate each other, then $w_{a,k} = 0$. Note that $w_{a,k}$ can be in between -1 and 1 inclusive.

CF is good for recommender systems because it is based on the ratings of other customers who have similar preferences. Although CF can be regarded as a good choice for a recommender system, there is still much more room for improvement in prediction quality. To do so, CF needs a more refined neighbor selection technique.

2.3 Clustering based CF

We need to use the clustering technique which is suitable for the numeric values because preferences are represented generally as numeric values which are rated by the customers in CF. So the k-Means clustering method is good for the clustering based CF.

The k-Means clustering method creates k clusters each of which consists of customers who have similar preferences among themselves. In this method we first select arbitrarily k customers as the initial center points of the k clusters, respectively. Then each customer is assigned to a cluster in such a way that the distance between the customer and the center of a cluster is minimized. The distance is calculated using the Euclidean distance, that is, a square root of the element-wise square of the difference between the customer and each center point.

We then calculate the mean of each cluster based on the customers who currently belong to the cluster. The mean is now considered as the new center of the cluster. After finding new centers, we compute the distance between the new center and each customer as before in order to find to which cluster the customer should belong. Recalculating the means and computing the distances are repeated until a terminating condition is met. The condition is in general how far each new center has moved from the previous center; that is, if all the new centers moved within a certain distance, we terminate the loop.

If the clustering process is terminated, we choose the cluster with the shortest Euclidean distance from its center to the test customer. Finally, prediction for the test customer is calculated with all the customers in the chosen cluster. The clustering based neighbor selection method can give a recommendation quickly to the customer in case of the very large-scale dataset, because it selects customers in the best cluster only as neighbors[2, 7].

3. Improving Prediction Quality in CF

We first describe a neighbor selection algorithm(NSA)[16]. It considers both high and low similarities with respect to the test customer and exploits the transitivity of similarity using a graph approach. We then explain refined NSA using weighted similarity and ranked clustering for expanding the candidate neighbors in CF.

3.1 NSA

The key ideas of NSA are to exploit the transitivity of similarities and to consider both higher and lower similarities in selecting neighbors. We regard a portion of the input dataset a complete undirected graph in which a vertex represents each customer and a weighted edge corresponds to the similarity between two end points(customers) of the edge.

NSA creates k clusters from the input dataset with the k-Means clustering method. Then it finds the best cluster C with respect to the test customer u among the k clusters. NSA searches the customers in C to find the 'best' vertex v with the highest similarity with respect to u. NSA then searches the vertices adjacent to v who have the similarities either larger than H or smaller than L, where H and L are some threshold values for the Pearson correlation coefficients. These thresholds can be

determined with various experiments. Note that as the threshold values change, so does the size of the neighbors. The search is performed in a breath first manner. That is, we search the adjacent vertices of v according to H and L to find the neighbors of u, and then search the adjacent vertices of each neighbor of v in turn. The search stops when we have enough neighbors for prediction.

Fig. 1 depicts an example of how NSA selects the neighbors of v who has the higher similarities with respect to u when the search depth is 2. A solid line indicates that the weight on the edge is larger than H. A dotted line means that the weight of the edge is smaller than L.

We now describe NSA in detail. In the algorithm, before we apply the graph approach to the input dataset, we want to remove insignificant customers with the k-means clustering method. After we obtain k clusters with the k-means clustering method, we choose a cluster(the *'best'* cluster) whose mean has the highest similarity with respect to the test customer u. We then apply the graph approach only to the best cluster. The following describes the overall algorithm in detail. When the algorithm is terminated, the set *Neighbors* is returned as output. The pseudo code for *NSA* is shown in Procedure1.

The Neighbor Selection Algorithm
Input: the test customer u, the input dataset S
Output: *Neighbors*

1. Create k clusters from S with the k Means clustering method;
2. Find the best cluster C for the test customer u;
3. Find the best customer v in cluster C who has the highest similarity with respect to u;
4. Add v to *Neighbors*;
5. Traverse C from v in a breadth first manner when visiting vertices(customers). The similarity of the customer is checked to see if it is either higher than H or lower than L. If so, add the customer to *Neighbors*;

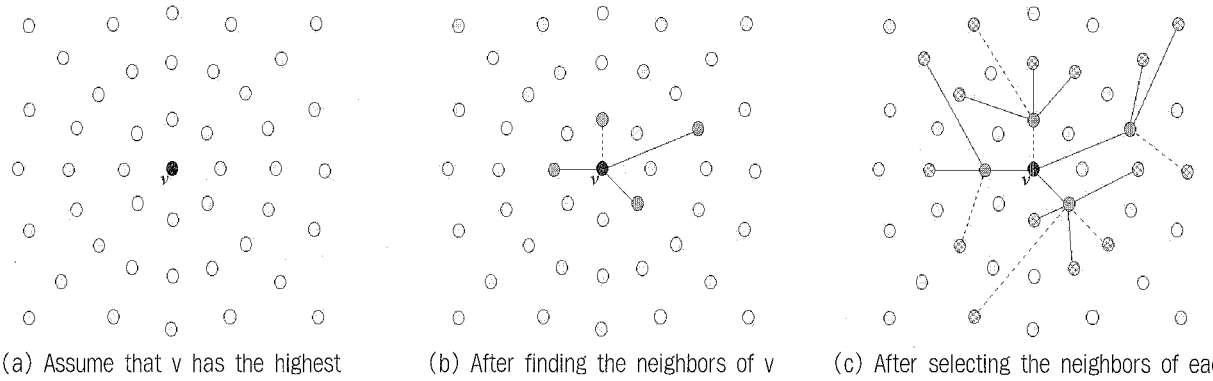
Note that Step5 is executed until we get enough neighbors and such terminating condition can be imposed with how many levels(depths) we search from v in a breadth first manner.

3.2 Refined Neighbor Selection Algorithm

We refine the searching done in NSA by associating a weight into the correlation coefficient between a pair of customers. The associated weight is applied to a similarity

Procedure 1 A Pseudo Code for NSA

```
1: Create k clusters with the k means clustering method;
2: Find the best cluster C for a given test customer u;
3: Unmark all the vertices in Cluster C;
4: d = 0; {d is for counting the search depth}
5: Find the neighbor v with the highest similarity with respect to u;
6: Mark v;
7: Insert (v, d) into Q;
   {Q holds temporarily the neighbors for a breadth first search}
8: Add v to Neighbors;
9: while (Q is not empty) do
10:  Delete (w,d) from Q;
11:  if (d < depth_count) then
     {depth_count controls the loop termination}
12:   for each unmarked neighbor x of w do
13:    if the weight similarity between x and w is either greater than H or
       less than L then
14:      Mark x;
15:      Insert (x,d+1) to Q;
16:      Add x to Neighbors;
17:    end if
18:   end for
19:  end if
20: end while
21: return Neighbors;
```



(a) Assume that v has the highest similarity for the test customer

(b) After finding the neighbors of v (the search depth=1)

(c) After selecting the neighbors of each neighbor of v (the search depth=2)

(Fig. 1) An example of searching the neighbors in NSA.

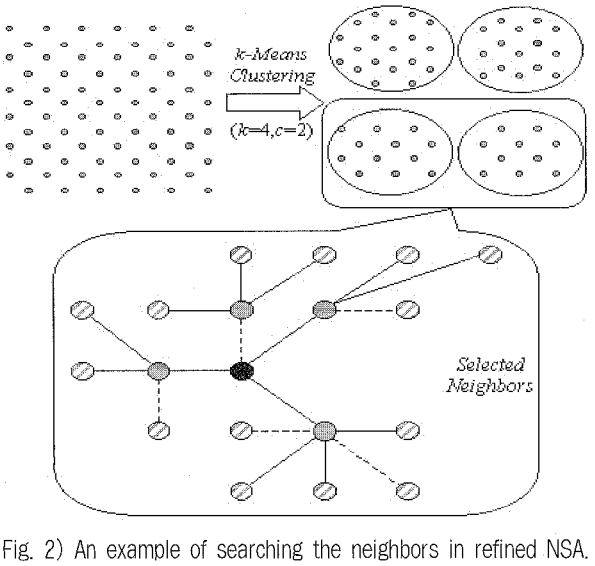
so that the closer vertices(customers) to the best vertex v have better chances to belong to the neighbors of u than others. Equation (2) is to calculate the weighted similarity $w'_{a,k}$ between customers a and k according to the search depth d in the graph.

$$w'_{a,k} = \begin{cases} w_{a,k} + \dfrac{|1 - w_{a,k}|}{\omega^d}, & w_{a,k} \geq \delta \\ w_{a,k} - \dfrac{|1 + w_{a,k}|}{\omega^d}, & w_{a,k} \leq -\delta \end{cases} \quad (2)$$

The new correlation coefficient $w'_{a,k}$ can be derived from the correlation coefficient $w_{a,k}$ in Equation (1). In the above equation, ω is a weight constant and $\omega>1$, d is a search depth, and δ is a threshold value. Observe that as d increases $|w'_{a,k}|$ gets larger. Note that $-1 \leq w_{a,k} \leq 1$.

NSA chooses only the best cluster among k clusters and searches the best cluster for finding the neighbors. However, the way we select the best cluster does not guarantee that there is no customer with higher similarity than H or lower similarity than L in other clusters. That is, there may be valuable customers with respect to the test customer in other clusters. In the proposed neighbor selection algorithm, we rank the k clusters created by the k-Means clustering method in terms of the similarity between the test customer and the mean of each cluster, and expand the search to the next cluster in the rank for finding the missing neighbors.

Fig. 2 shows how the proposed neighbor selection algorithm selects the neighbors. First, we rank the clusters in terms of similarities. In this figure we assume there are k=4 clusters and we search the best two clusters(c = 2).



(Fig. 2) An example of searching the neighbors in refined NSA.

And we search the neighbors in these two clusters as NSA does, yet refine the search with the concept of the weighted similarity.

3.3 The Computational Complexity

The k-eans clustering algorithm is arguably the most well-nown clustering algorithm in use today. Its computational complexity is O(krn), where k is the number of desired clusters, r is the number of iterations, and n is the size of the dataset[15]. In collaborative filtering the complexity for the prediction depends on the size of the neighbors. Therefore, if we consider all the points (customers) in the chosen cluster (the best cluster) as the neighbors, the complexity is only O(c), where c is the size of the chosen cluster.

The complexity of the breadth-irst search (BFS) in a complete undirected graph is O(n^2), where n is the size of the vertex set. NSA searches the neighbors with transitivity in a graph using BFS, stops the search if the search depth is greater than a threshold value predetermined. BFS for the neighbors using the transitivity mechanism in a complete undirected graph takes O(dn), where d is the search depth. If we consider only the chosen cluster, the search complexity is O(dc).

Although *NSA* searches the neighbors using BFS, it searches only 'not selected' vertices, that is, all vertices selected as the neighbors in the previous search steps are considered as the selected vertices. Therefore, the complexity T(c) of *NSA* can be computed with (3). T(c) is derived from $\sum_d T_d = T_1 + T_2 + \cdots + T_d$, where T_d is the complexity when the search depth is d. $T(c)$ is computed by each step of the following appendix. In (3) and the appendix S_d is the size of the selected vertices when the search depth is d and S_d^i s the size of the selected vertices for each vertex i of S_{d-1}. For each depth if there is only one selected vertex, then $T(c)$ is O(dc), which is the worst case. If all vertices are selected when the search depth is one, then T(c) is O(c), which is the best case.

$$T(c) = \sum_d (S_{d-1}(c - \sum_{i=1}^{d-1} S_i)) - \sum_d (\sum_{i=1}^{S_{d-1}-1} (S_{d-1} - i) S_d^i),$$
$$where \;\; S_0 = 1, \sum_d S_d = c, and \sum_i S_d^i = S_d. \quad (3)$$

4. Experimental Results

4.1 Experimental Dataset

In the experiments we used the EachMovie dataset of the Compaq Computer Corporation for evaluating prediction quality of the proposed recommendation system[6]. The

EachMovie dataset consists of 2,811,983 preferences for 1,628 movies rated by 72,916 customers explicitly. The customer preferences are represented as numeric values from 0 to 1 at an interval of 0.2, i.e., 0, 0.2, 0.4, 0.6, 0.8, and 1.

For the experiment, we retrieved 3,763 customers who rated at least 100 movies among all the customers in the dataset. We have chosen randomly fifty customers as the test customers and the rest of the customers are the training customers. For each test customer, we chose five movies randomly that are actually rated by the test customer as the test movies. The final experimental results are averaged over the results of the test sets.

4.2 Experimental Metrics

One of the statistical prediction accuracy metrics for evaluating a recommender system is the mean absolute error(MAE) which is the mean of the errors of the actual customer ratings against the predicted ratings in an individual prediction[1, 8, 10, 11, 12]. MAE is computed by Equation (4). In the equation, N is the total number of predictions and ε_i is the error between the predicted rating and the actual rating for item i. The lower MAE is, the more accurate prediction with respect to the numerical ratings of customers we get.

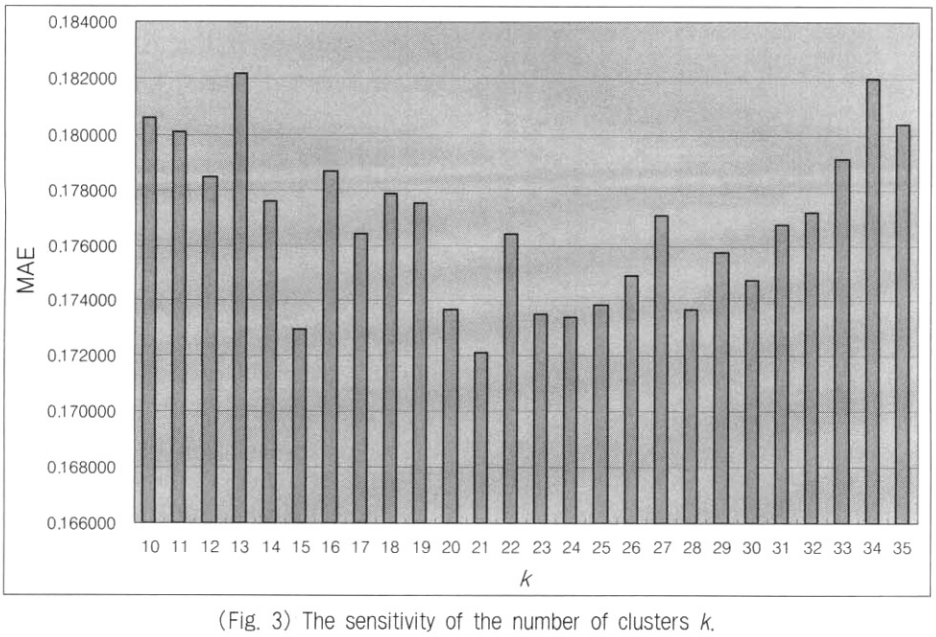
$$|E| = \frac{\sum_{i=0}^{N} |\varepsilon_i|}{N}. \quad (4)$$

4.3 Experimental Results

For comparing each method, we have implemented five recommendation systems for the experiments. The first one is the recommendation system only with the clustering-based CF, called *KCF*. The second one is the proposed neighbor selection based on *KCF*, called *NSA*. The third one is *NSA* with weighted similarity, called *NSA-w*. The forth one is *NSA* with ranked clustering, called *NSA-o*. And the last one is *NSA* with both weighted similarity and ranked clustering and is called *NSA-wo*.

The experimental results are shown in Table 1 and Table 2. We determined the value of k for the clustering-based method through various experiments. That is, this value, 21, for k gave us the smallest MAE among others as shown in Fig.3. MAE values tend to be decreased gradually up to k=21, and then be increased thereafter. Hence we decided k=21. Among the parameters in the table, L and H denote that the threshold values for the Pearson correlation coefficients. It is better to select H and L that $|H| < |L|$. For example, if L = -0.8 and H = 0.7, then we select a neighbor whose similarity is either smaller than -0.8 or larger than 0.7. In the table ω denotes a weight constant, ω is a threshold value, c is the number of clusters, and d is a search depth. We can select that the value of d with the lower value for the small search space.

The results in the table show that the proposed systems outperform the clustering-based system. The prediction improvement ratio of each proposed system to *KCF* is more than three or four percent. The improvement ratio of *NSA* and *NSA-w* to *KCF* are about 3.7%. *NSA-o* has



(Fig. 3) The sensitivity of the number of clusters k.

〈Table 1〉 Experimental results.

Methods	MAE	Parameters
KCF	0.172146	k = 21
NSA	0.165700	k = 21, H = 0.7, L = -0.8, d = 2
NSA-w	0.165675	k = 21, H = 0.7, L = -0.8, ω = 2, δ = 0.8, d = 2
NSA-o	0.164841	k = 21, H = 0.5, L = -0.6, d = 1, c = 2
NSA-wo	0.164775	k = 21, H = 0.5, L = -0.6, ω = 2, δ = 0.8, d = 1, c = 2

〈Table 2〉 Experimental results for various cases.

Methods	MAE	Parameters
NSA	0.166095	k = 21, H = 0.4, L = -0.6, d = 1
	0.165700	k = 21, H = 0.7, L = -0.8, d = 2
	0.167418	k = 21, H = 0.7, L = -1.0, d = 3
NSA-o	0.164841	k = 21, H=0.5, L = -0.6, c = 2, d = 1
	0.166669	k = 21, H = 0.7, L = -1.0, c = 2, d = 2
	0.171530	k = 21, H = 0.7, L = -1.0, c = 2, d = 3
	0.167479	k = 21, H = 0.5, L = -0.8, c = 3, d = 1
	0.166952	k = 21, H = 0.7, L = -1.0, c = 3, d = 2
	0.166580	k = 21, H = 0.8, L = -1.0, c = 3, d = 3

improved in prediction quality and the improvement ratio is about 4.2%. And *NSA-wo* has the best prediction quality among others and the improvement ratio is about 4.3%.

The results in the table show that *NSA-w* outperforms *KCF* and *NSA* when the search depth is 2. This fact shows that the weighted similarity concept helps selecting the refined neighbors in *NSA*. We found that when the search depth is greater than 2, neither *NSA* or *NSA-w* did not provide better results, because as the size of the neighbors gets larger the chances that unnecessary customers are included in the neighbors get higher. Table 1 also shows that *NSA-o* which expands the search space through cluster ranking finds valuable missing neighbors in the clusters other than the best cluster. When we expand *NSA* with both weighted similarities and cluster ranking, we could achieve the best prediction quality.

5. Conclusion

It is crucial for a recommender system to have the capability of making accurate prediction by retrieving and analyzing customers' preferences. Because CF is widely used for recommender systems, various efforts to overcome its drawbacks have been made to improve prediction quality.

It is important to select neighbors properly in order to make up the weak points in collaborative filtering and to improve prediction quality. In this paper we proposed a refined neighbor selection algorithm that finds meaningful neighbors using weighted similarity and ranked clustering based on a graph approach along with clustering. The proposed methods utilize a clustering technique and transitivity mechanism. The methods consider both high and low similarities of each customer to give valuable information in prediction. To improve prediction quality the methods also search with concepts of weighted similarity and ranked clustering.

The experimental results show the proposed recommender system selects meaningful neighbors for the high prediction quality and then provides the better prediction quality than others. Therefore the clustering-based recommender systems using refined neighbor selection methods can be a choice to solve the very large scale dataset problem with high prediction quality.

The various datasets have in general domain dependent characteristics in the real world. So the recommender systems shown in this paper can be restricted to the special domain datasets such as the movies, musical CDs.

Acknowledgements

We thank the Compaq Equipment Corporation for permitting us to use the EachMovie dataset. We thank the anonymous reviews for their many suggestions for improving this paper. This work has been supported by the BK21 Research Center for Intelligent Mobile Software at Yonsei University in Korea.

References

[1] B. M. Sarwar, G. Karypis, J. A. Konstan, J. T. Riedle, "Application of Dimensionality Reduction in Recommender System - A Case Study," Proceedings of the ACM WebKDD 2000 Web Mining for E-Commerce Workshop, 2000.

[2] B.M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender Systems for Large Scale E-Commerce: Scalable Neighborhood Formation Using Clustering," Proceedings of the Fifth International Conference on Computer and Information Technology, 2002.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of Recommendation Algorithms for E-Commerce," Proceedings of the ACM E-Commerce 2000 Conference, 2000.

[4] Batul J. Mirza, Benjamin, J. Keller, "Studying Recommendation Algorithm by Graph Analysis," Journal of Intelligent Information Systems, Vol.20, No.2, pp.131-160, 2003.

[5] Cosley, D., Lam, S.K., Albert, I., Konstan, J., and Riedl, J., "Is Seeing Believing? How Recommender Systems Influence Users' Opinions," Proceedings of CHI2003 Conference on Human Factors in Computing Systems, pp.585-592, 2003.

[6] EachMovie Collaborative Filtering Data Set. Compaq

Computer Corporation, http://www.research.compaq.com/SRC/eachmovie/

[7] G. Xue, C. Lin, and Q.E. Yang, "Scalable Collaborative Filtering Using Cluster-based Smoothing," Proceedings of the ACM SIGIR Conference, pp.114 121, 2005.

[8] J. Herlocker, J. Konstan, L. Terveen, and J. Riedle, "Evaluating Collaborative Filtering Recommender Systems," ACM Transactions on Information Systems, Vol.22, No.1, pp.5-53, 2004.

[9] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying Collaborative Filtering to Usenet News," Communications of the ACM 40, pp.77 - 87, 1997.

[10] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999.

[11] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proceedings of the Conference on Uncertainty in Artificial Intelligence, pp.43-52, 1998.

[12] M. O'Connor, and J. Herlocker, "Clustering Items for Collaborative Filtering," Proceedings of the ACM SIGIR Workshop on Recommender Systems, 1999.

[13] N.Yamamoto, M. Saito, M. Miyazaki, H. Koike, "Recommendation Algorithm Focused on Individual Viewpoints," Proceedings of the Conference on CCNC, 2005.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proceedings of the ACM CSCW94 Conference on Computer Supported Cooperative Work, pp. 175-186, 1994.

[15] S. Kantabutra, "Cluster computing as a tool in theoretical computer science," Proceedings of the National Workshop on Cluster Computing 2003, pp.1-16, 2003.

[16] T.-H. Kim, S.-B. Yang, "An Improved Neighbor Selection Algorithm in Collaborative Filtering," IEICE Transactions on Information and Systems, Vol.E88-D, No.5, pp.1072-1076, 2005.

김 택 헌

e-mail : kimthun@cs.yonsei.ac.kr

1996년 동국대학교 전자계산학과 (공학사)

2000년 연세대학교 컴퓨터과학과 (공학석사)

2005년 연세대학교 컴퓨터과학과 (공학박사)

1996년~2000년 삼성SDS 정보기술연구소 연구원

2005년~2006년 연세대학교 컴퓨터과학과 BK21 박사후연구원

2006년~현 재 연세대학교 컴퓨터과학과 BK21 연구교수

관심분야 : Information Searching, Filtering & Retrieval, Recommender Systems, Personalization

양 성 봉

e-mail : yang@cs.yonsei.ac.kr

1981년 연세대학교 (공학사)

1984년 University of Oklahoma 컴퓨터과학 (공학석사)

1992년 University of Oklahoma 컴퓨터과학 (공학박사)

1993년~1994년 전주대학교 전자계산학과 전임강사

1994년~현 재 연세대학교 컴퓨터과학과 교수

관심분야 : Information Filtering & Retrieval, Mobile Computing, 3D Graphics, Computer Theory