# Using Attributes to Improve Prediction Quality in Collaborative Filtering

Taek-Hun Kim and Sung-Bong Yang

Dept. of Computer Science, Yonsei University
Seoul, 120-749, Korea
{kimthun,yang}@cs.yonsei.ac.kr

**Abstract.** To save customers' time and efforts in searching the goods in the Internet, a customized recommender system is required. It is very important for a recommender system to predict accurately by analyzing customer's preferences. 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. Because a recommender system using collaborative filtering predicts customer's preferences based only on the items without useful information on the attributes of each item, it may not give high quality recommendation consistently to the customers. In this paper we show that exploiting the attributes of each item improves prediction quality. We analyze the dataset and retrieve the preferences for the attributes because they have not been rated by customers explicitly. In the experiment the MovieLens dataset of the GroupLens Research Center has been used. The results on various experiments using several neighbor selection methods which are quite popular techniques for recommender systems show that the recommender systems using the attributes provide better prediction qualities than the systems without using the attribute information. Each of the systems using the attributes has improved the prediction quality more than 9%, compared with its counterpart. And the clustering-based recommender systems using the attributes can solve the very large-scale dataset problem without deteriorating prediction quality.

## 1 Introduction

Nowadays, customers spend so much time and efforts in finding the best suitable goods since more and more information is placed on-line. To save their time and efforts in searching the goods they want, a customized recommender system is required. A customized recommender system should predict the most suitable goods for customers by retrieving and analyzing their preferences. A recommender system utilizes in general an information filtering technique called collaborative filtering which is widely used for such recommender systems as Amazon.com and CDNow.com[1][2][3][4]. A recommender system using collaborative filtering, we call it *CF*, is based on the ratings of the customers who have similar preferences with respect to a given (test) customer.

CF calculates the similarity between the test customer and each of other customers who have rated the items that are already rated by the test customer. CF uses in general the Pearson correlation coefficient for calculating the similarity, but it assumes that there must exist at least one item which has already been rated by both the test customer and one of the other customers. A weak point of the "pure" CF is that it uses all other customers including "useless" customers as the neighbors of 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. Another weak point of CF is that it never considers customer's preferences on the attributes of each item.

There have been many investigations to select 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[1][4][6][7][10]. These techniques then predict customer's preferences for the items based on the results of the neighbors' evaluation on the same items.

The recommender system with the clustering-based neighbor selection method is known to predict with worse accuracy than the systems with other neighbor selection methods, but it can solve the very large-scale problem in recommender systems. With millions of customers and items, a recommender system running existing algorithms will suffer serious scalability problem. So it is needed a new idea that can quickly produce high prediction quality, even for the very large-scale problem[8][10].

CF works quite well in general, because it is based on the ratings of other customers who have similar preferences. However, CF may not provide high quality recommendations for the test customer consistently, because it does not consider the attributes of each item and because it depends only on the ratings of other customers who rated the items that are already rated by the test customer. To improve prediction quality, CF needs reinforcements such as utilizing "useful" attributes of the items and using a more refined neighbor selection method. Item attributes mean its originality of the item it holds which differs from the others. In general, they can be obtained through the properties of the goods in real world.

In this paper we show that exploiting the attributes of each item improves prediction quality. We analyze the dataset and retrieve the preferences for the attributes, because they have not been rated by the customers explicitly. In the experiment the MovieLens dataset of the GroupLens Research Center has been used[11]. The dataset consists of 100,000 preferences for 1,682 movies rated by 943 customers explicitly.

We show the impact of utilizing the attributes information on the prediction qualities through various experiments. The experimental results show that the recommender systems using the attribute information provide better prediction qualities than other methods that do not utilize the attributes. Each of the systems using the attributes has improved the prediction quality more than 9%, compared with its counterpart. And besides the clustering-based CF using the attributes can solve the very large-scale problem without deteriorating prediction quality.

The rest of this paper is organized as follows. Section 2 describes briefly CF and several neighbor selection methods. Section 3 illustrates how to utilize the attributes of items for recommender systems in detail. In Section 4, the experimental results are presented. Finally, the conclusions are given in Section 5.

## 2 Collaborative Filtering and Neighbor Selection Methods

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 the customers. Predicting the 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, called *neighbors*, 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 as in Equation (2) [2][3][4][6][8].

$$P_{a,i} = \overline{r_a} + \frac{\sum_k \{w_{a,k} \times (r_{k,i} - \overline{r_k})\}}{\sum_k \mid w_{a,k} \mid} \ . \tag{1}$$

$$w_{a,k} = \frac{\sum_j (r_{a,j} - \overline{r_a})(r_{k,j} - \overline{r_k})}{\sqrt{\sum_j (r_{a,j} - \overline{r_a})^2 \sum_j (r_{k,j} - \overline{r_k})^2}} \ . \tag{2}$$

In the above equations $P_{a,i}$ is the preference of customer $a$ with respect to item $i$. $\overline{r_a}$ and $\overline{r_k}$ 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, $w_{a,k} > 0$. We denote $\mid w_{a,k} \mid$ to indicate how much customer $a$ tends to agree with customer $k$ on the items that both customers have already rated. In this case, customer $a$ is a "positive" neighbor with respect to customer $k$, and vice versa. If they have opposite ratings for an item, then $w_{a,k} < 0$. Similarly, customer $a$ is a "negative" neighbor with respect to customer $k$, and vice versa. In this case $\mid w_{a,k} \mid$ 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.

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 reinforcements such as utilizing "useful" attributes of the items as well as a more refined neighbor selection. In the rest of this section we describe several neighbor selection methods.

## 2.1 The $k$-Nearest Neighbor Selection

The $k$-nearest neighbor method selects the nearest $k$ neighbors who have similar preferences to the test customer by computing the similarities based on their preferences. It only uses the $k$ neighbors who have higher correlation with the test customer than others, while CF suffers from considering all the customers in the input dataset for calculating the preference of the test customer. Thus CF should even consider some customers who may give bad influences on prediction quality. It has been shown in several investigations that the recommender system with the $k$-nearest neighbor selection method has better quality of prediction than the "pure" CF[1][4][6].

## 2.2 The Threshold-based Neighbor Selection

The threshold-based neighbor selection method selects the neighbors who belong to a certain range with respect to the similarities of the preferences. Contrary to the $k$-nearest neighbor selection method, the number of neighbors selected by this method varies, because it selects neighbors according to a certain threshold value $\tau$.

In the recommender system with the threshold-based neighbor selection, the positive neighbors whose correlations to the test customer are greater than and equal to $\tau$ are selected as the neighbors[6]. However, it is also needed that we include the "negative" neighbors whose correlations to the test customer are less than and equal to $\tau$, because they could contribute toward the better neighbor selection for the test customer as they provide negative "opinions" to the test customer. It is obvious that selecting only negative neighbors results in worse prediction qualities than the case in which only positive neighbors are selected, intuitively.

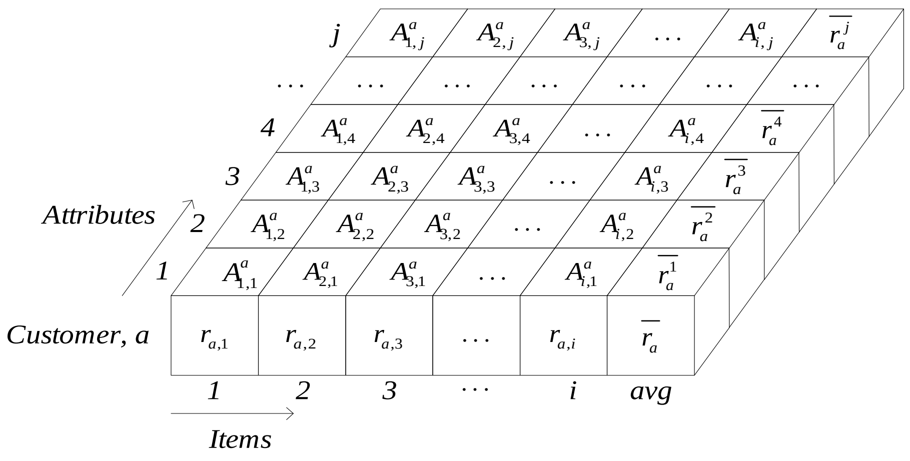
## 2.3 The Clustering-based Neighbor Selection

The $k$-means clustering method creates $k$ clusters each of which consists of the customers who have similar preferences among themselves. In this method we first select $k$ customers arbitrarily 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. The Pearson correlation coefficient can be substituted for the Euclidean distance.

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 the new center of each cluster, we compute the distance between the new center and each customer as before in order to find the cluster to which 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.

## 3 Using Attributes to Improve Prediction Quality



Attributes
Customer, a
Items
$j$
$A^a_{1,j}$ $A^a_{2,j}$ $A^a_{3,j}$ ... $A^a_{i,j}$ $\overline{r^j_a}$
4
$A^a_{1,4}$ $A^a_{2,4}$ $A^a_{3,4}$ ... $A^a_{i,4}$ $\overline{r^4_a}$
3
$A^a_{1,3}$ $A^a_{2,3}$ $A^a_{3,3}$ ... $A^a_{i,3}$ $\overline{r^3_a}$
2
$A^a_{1,2}$ $A^a_{2,2}$ $A^a_{3,2}$ ... $A^a_{i,2}$ $\overline{r^2_a}$
1
$A^a_{1,1}$ $A^a_{2,1}$ $A^a_{3,1}$ ... $A^a_{i,1}$ $\overline{r^1_a}$
$r_{a,1}$ $r_{a,2}$ $r_{a,3}$ ... $r_{a,i}$ $\overline{r_a}$
1 2 3 ... i avg

**Fig. 1.** The item-attributes matrix for a customer

In order to utilize the attributes of the items in the process of prediction, we consider a matrix of items-attributes for a customer as in Fig. 1. This figure shows the information of the items-attributes for customer $a$. In this figure $r_{a,i}$ denotes customer $a$'s rating for item $i$. Note that $r_{a,i} \in \{-, 1, 2, 3, 4, 5\}$, where '-' indicates "no rating". An item may have an arbitrary combination among $j$ attribute values as its attribute values; $\mathit{Attribute}(i) \subset \{1, 2, 3, \ldots, j\}$. Note that $\mathit{Attribute}(i) \neq \emptyset$. And $A^a_{i,j} \in \{0, 1\}$.

$$P_{a,i} = A(\overline{r_{a,i}}) + \frac{\sum\limits_k \{w_{a,k} \times (r_{k,i} - A(\overline{r_{k,i}}))\}}{\sum\limits_k \mid w_{a,k} \mid} \; . \tag{3}$$

$$A(\overline{r_{a,i}}) = \frac{\sum_{j}\{A^{a}_{i,j} \times \overline{r^{j}_{a}}\}}{N_{a,i}} \quad , \quad \overline{r^{j}_{a}} = \frac{\sum_{i}\{A^{a}_{i,j} \times r_{a,i}\}}{M_{a,j}} \quad . \tag{4}$$

$$A(\overline{r_{k,i}}) = \frac{\sum_{j}\{A^{k}_{i,j} \times \overline{r^{j}_{k}}\}}{N_{k,i}} \quad , \quad \overline{r^{j}_{k}} = \frac{\sum_{i}\{A^{k}_{i,j} \times r_{k,i}\}}{M_{k,j}} \quad . \tag{5}$$

We propose Equation (3) as a new prediction formula in order to predict customer's preferences. In this equation, $A(\overline{r_{a,i}})$ and $A(\overline{r_{k,i}})$ in Equations (4) and (5) are the averages of customer $a$ and $k$'s attribute values, respectively. In each equation, $N_{a,i}$ and $N_{k,i}$ is the number of "valid" attributes for item $i$ and $M_{a,j}$ and $M_{k,j}$ is the number of "valid" items that has attribute $j$ for $a$ customer, respectively. The valid attribute means $A^{a}_{i,j} = 1$ and $A^{k}_{i,j} = 1$ and the valid item means $r_{a,i}$ and $r_{k,i}$ is not "-", respectively. All other terms in the above equations are the same terms defined for the previous equations.

There are a lot of items which have different attributes for the item for new prediction. Therefore, if we retrieve the attributes of the items more accurately and use them to the prediction process with Equation (3), then we can get more accurate prediction quality than the case without considering the attributes.

## 4 Experimental Results

### 4.1 Experiment Environment

In order to evaluate the prediction accuracy of the proposed recommendation system, we used the MovieLens dataset of the GroupLens Research Center [11]. The dataset consists of 100,000 preferences for 1,682 movies rated by 943 customers explicitly. The customer preferences are represented as numeric values from 1 to 5 at an interval of 1, that is, 1, 2, 3, 4, and 5. A higher value means higher preference. In the MovieLens dataset, one of the valuable attributes of an item is "the genre" of a movie. There are nineteen different genres as shown in Table 1.

**Table 1.** The genre attributes of a movie

| | | | | |
|---|---|---|---|---|
| Unknown | Action | Adventure | Animation | Children's |
| Comedy | Crime | Documentary | Drama | Fantasy |
| Film-Noir | Horror | Musical | Mystery | Romance |
| Sci-Fi | Thriller | War | Western | |

For the experiment, we have chosen randomly 10% of customers out of all the customers in the dataset as the test customers. The rest of the customers are regarded as the training customers. For each test customer, we chose ten movies randomly that are actually rated by the test customer as the test movies. The final experimental results are averaged over the results of five different test sets for a statistical significance.

## 4.2 Experimental Metrics

One of the statistical prediction accuracy metrics for evaluating a recommender system is the mean absolute error (MAE). MAE is the mean of the errors of the actual customer ratings against the predicted ratings in an individual prediction [1][6][7][8]. MAE can be computed with Equation (6). In the equation, $N$ is the total number of predictions and $\varepsilon_i$ 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.

$$\mid E \mid = \frac{\sum_{i=0}^{N} \mid \varepsilon_i \mid}{N} \quad . \tag{6}$$

## 4.3 Experimental Results

We compared the recommendation systems using the attribute information with those without using them. We have implemented three recommender systems each of which uses different neighbor selection method. The first system is CF with the $k$-nearest neighbor selection method ($KNCF$). The second system is CF with the threshold-based neighbor selection ($TNCF$). The third one is CF with the k-means clustering ($KMCF$).

For TNCF, three different systems have implemented. The first system is TNCF with positive neighbors ($TNCF\_P$). The second one is TNCF with negative neighbors ($TNCF\_N$). And the last one is TNCF with both positive and negative neighbors ($TNCF\_A$).

For KMCF, we also have two different settings. One is KMCF with the Euclidean distance as a distance ($KMCF\_E$). And the other is KMCF with the Pearson correlation coefficient as a distance ($KMCF\_C$).

The experimental results are shown in Table 2. We determined the parameters which gave us the smallest MAE through various experiments. The value of $k$ for KNCF is the number of neighbors with $k$ highest correlation. Among the parameters in TNCF, positive and negative values denote that the $\tau$ values for the Pearson correlation coefficients. For example, if the positive vale is 0.2 and the negative value is -0.1, then we select a neighbor whose similarity is either smaller than and equal to -0.1 or larger than and equal to 0.2. If there is a single parameter, then we select neighbors whose similarities with respect to the value only. And the last the value of $k$ for KMCF is the number of clusters.

As shown in Table 2, each system has been tested both with and without utilizing the attributes. The results in the table show that the systems using the

**Table 2.** The experimental results

| Recommender systems | MAE's | Parameters | Attributes | Improvement ratios |
|---|---|---|---|---|
| KNCF | 0.686170 | $k$=130 | Used | 9.86% |
| | 0.761210 | $k$=120 | Not used | |
| TNCF_P | 0.689162 | $\tau$=0.2 | Used | 9.41% |
| | 0.760716 | $\tau$=0.2 | Not used | |
| TNCF_N | 0.745659 | $\tau$=-1.0 | Used | 11.02% |
| | 0.838030 | $\tau$=-1.0 | Not used | |
| TNCF_A | 0.681760 | $\tau$=(0.2,-0.1) | Used | 9.66% |
| | 0.754619 | $\tau$=(0.2,-0.1) | Not used | |
| KMCF_E | 0.744384 | $k$=2 | Used | 3.88% |
| | 0.774412 | $k$=2 | Not used | |
| KMCF_C | 0.729218 | $k$=2 | Used | 4.41% |
| | 0.762834 | $k$=2 | Not used | |

attributes outperform those without considering them. The prediction accuracy improvement ratio of each system using the attributes to the one without considering them is more than 9% except for the clustering systems, KMCF_E and KMCF_C.

**Table 3.** The usage of the attributes in prediction

| Cases | A | B | C | D |
|---|---|---|---|---|
| Used equations | (4) and (5) | (4) only | (5) only | none |

Table 3 shows four cases of combination according to whether Equation (4) or (5) is applied to Equation (3) or not. If Equation (4) or (5) is not used, then the item corresponded to Equation (1) is substituted. Table 4 shows the experimental results on these cases.

The experimental results in Table 4 show us that the prediction with case A that the attributes are applied to both the test customer and the neighbors, provides the best prediction accuracy in KNCF and TNCF. And in these methods the prediction with case B also has as good prediction quality as the prediction with case A. On the other hand KMCF provides the best prediction when we use the case B in both KMCF_E and KMCF_C. The prediction accuracy improvement ratio of KMCF with case B to the case D is more than 9%.

In the results we found the TNCF_A (case A) is the best prediction quality among others. And we can see the fact that the prediction with case A through C provides more prediction accuracy than the one with case D which never considers the attributes of an item. And the last the KMCF_C (case B) provides as good as TNCF_A in prediction quality. This fact means that the clustering-based CF can solve the very large-scale problem without deteriorating prediction quality.

**Table 4.** The experimental results(for four cases)

| Methods | MAE | Parameters | Types |
|---|---|---|---|
| KNCF | 0.686170 | $k$=130 | A |
| | 0.692809 | $k$=200 | B |
| | 0.767625 | $k$=120 | C |
| | 0.761210 | $k$=120 | D |
| TNCF_P | 0.689162 | $\tau$=0.2 | A |
| | 0.696775 | $\tau$=0.2 | B |
| | 0.767223 | $\tau$=0.2 | C |
| | 0.760716 | $\tau$=0.2 | D |
| TNCF_N | 0.745659 | $\tau$=-1.0 | A |
| | 0.745667 | $\tau$=-1.0 | B |
| | 0.838022 | $\tau$=-1.0 | C |
| | 0.838030 | $\tau$=-1.0 | D |
| TNCF_A | 0.681760 | $\tau$=(0.2,-0.1) | A |
| | 0.687949 | $\tau$=(0.2,-0.1) | B |
| | 0.761572 | $\tau$=(0.2,-0.1) | C |
| | 0.754619 | $\tau$=(0.2,-0.1) | D |
| KMCF_E | 0.744384 | $k$=2 | A |
| | 0.704240 | $k$=2 | B |
| | 0.811596 | $k$=2 | C |
| | 0.774412 | $k$=2 | D |
| KMCF_C | 0.729218 | $k$=2 | A |
| | 0.692414 | $k$=2 | B |
| | 0.796347 | $k$=2 | C |
| | 0.762834 | $k$=2 | D |

## 5 Conclusions

It is very crucial for a recommender system to have a capability of making accurate prediction by retrieving and analyzing of customer's preferences. Collaborative filtering is widely used for recommender systems. Hence 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 showed that the recommender systems that exploit the attributes of each item indeed improve prediction qualities. The experimental results show the recommender systems have improved the prediction qualities more than 9%. And besides the clustering-based CF using the attributes can solve the large-scale problem without deteriorating prediction quality.

## Acknowledgements

We thank the GroupLens Research Center for permitting us to use the MovieLens dataset. This work was supported by the Korea Sanhak Foundation and the Brain Korea 21 Project in 2004.

## References

1. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John 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. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, Vol. 40. (1997) 77-87
3. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Proceedings of the ACM CSCW94 Conference on Computer Supported Cooperative Work. (1994) 175-186
4. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John T. Riedl: Analysis of Recommendation Algorithms for E-Commerce. Proceedings of the ACM E-Commerce 2000 Conference. (2000)
5. Basu, C., Hirsh, H., and Cohen, W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. Proceedings of the AAAI. (1998) 714-720
6. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl: An Algorithmic Framework for Performing Collaborative Filtering. Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval. (1999)
7. O'Connor M., and Herlocker J.: Clustering Items for Collaborative Filtering. Proceedings of the ACM SIGIR Workshop on Recommender Systems. (1999)
8. John S. Breese, David Heckerman, and Carl Kadie: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proceedings of the Conference on Uncertainty in Artificial Intelligence. (1998) 43-52
9. J. Benschafer, Joseph Konstan and John Riedl: Recommender Systems in E-Commerce. Proceedings of the ACM Conference on Electronic Commerce. (1999)
10. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John T. Riedle: Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. Proceedings of the Fifth International Conference on Computer and Information Technology. (2002)
11. MovieLens dataset, GroupLens Research Center, url: http://www.grouplens.org/.