Improving Recommender Systems by Incorporating Similarity, Trust and Reputation

ChanChamnab Than and SangYong Han*
Chung-Ang University, Seoul, Republic of Korea
tcchamnab@gmail.com, hansy@cau.ac.kr

Abstract

Recommender systems using traditional collaborative filtering suffer from some significant weaknesses, such as data sparseness and scalability. In this study, we propose a method that can improve the recommender systems by combining similarity, trust and reputation. We modify the way that neighbors are selected by introducing the trust and reputation metrics in order to develop new relations between users so that it can increase the connectivity and alleviate the data sparseness problem. Throughout our 2 different scenarios of experiment simulations conducted on MovieLens dataset and the comparison of our results with other trust-based collaborative filtering research, we found out that our proposed method outperforms for better recommendations in an effective way.

Keywords: Similarity, Trust, Reputation, Recommender System

1 Introduction

With the massive growth of internet and the emergence of E-commerce over the last decades, the web applications on World-Wide-Web these days can lead to information overload problem. The process of searching for relevant products or contents within an endless number of web pages to meet the desired items is taking quite a time and as the result, it is easy to make users fade away from the system. Recommender system [18] has shown to be an important and suitable choice to deal with this issue. The main role of recommender system is to assist users by providing a list of recommendations that might interest them based on their preferences and behaviors.

To date, a variety of recommender systems have been developed [7]. To our best knowledge: content-based, collaborative filtering and hybrid systems are the most three popular techniques that have been used to generate the recommendations. And the approach that has received the most attention is collaborative filtering. However, recommender systems using this approach commonly suffer from several limitations. A major problem in traditional collaborative filtering [13] is data sparseness. In the real-world web application where there will consist of millions of products and millions of items, a typical user tends to rate a small number of items. It is unlikely for two random users to rate some items in common. Therefore, similarity values in not computable in most cases.

The data sparseness [2] problem is particularly evident in cold-start problem. It refers to a new user or users who have rated few items. In these cases, it is difficult and impossible to provide a recommendation base on the little knowledge of profile similarity. This is a significant issue, since it is important to provide good recommendations to new users in order to retain them as customers.

Scalability [2] is another weakness in classic collaborative filtering. The basic for making accurate recommendations relies on finding similar users. A typical algorithm must compute a similarity values

Journal of Internet Services and Information Security (JISIS), volume: 4, number: 1, pp. 64-76
*Corresponding author: Department of Computer Science and Engineering, Chung-Ang University, Heukseok 1(il)-dong, Dongjak-gu, Seoul, Republic of Korea, Tel: +82-(0)28205327

for each pair of users. The computation overhead will quickly increase as the number of users increases, which leading to poor scalability.

The goal this research is to improve the accuracy and quality of recommendation that the traditional collaborative filtering provides. The proposed model will modify the way the neighbors are selected and weighted during the recommendation process. By using trust and reputation, we have developed a new relation between the users so that it can increase the connectivity of users and deal with the data sparseness problem.

The remainder of this paper is organized as follows: In Section 2 we discuss the existing literature reviews related to our research. In Section 3 we discuss the factors that contribute to our proposed method. Section 4 describes the experiments and their results. We conclude our research in Section 5 and give a guideline for future directions in Section 6.

2 Related Works

2.1 Recommender System

Recommender systems can help users to identify the items that suitable for their needs and preferences in an effective way. They are usually used to solve information overload problems and to increase sales in E-commerce mechanism [19].

Imagine that if a web application can offer to a user an automated and intelligent system which generates personalized recommendations, it will definitely increase the competitive advantage. The recommender system can range from online-shopping (books, movies, music, food, etc...), to a suggesting course for a student, or a particular kind of medical care for a patient. Recommender systems user various types of information to generate a recommendation, such as past purchase record, user profile, explicating ratings of items, or social network information.

A variety of recommendation techniques has been developed: content-based, collaborative filtering and hybrid recommendation systems. Content-based systems [4] analyze the characteristics of items previously rated by a user and generating a profile for a user based on the content descriptions of these items. These systems rely on information such as genre, actors, directors, etc. and match this against the learned preferences of the user in order to select recommendations. Lee et al.[11] propose a content-based online product recommendation using Amazon's book rating and review data to support business-to-consumer e-commerce.

Collaborative filtering systems [20] identify similar users and analyze their preferences to generate recommendation. The main characteristic of collaborative filtering is that is accumulates the user's ratings of products, identifies user with common ratings, and offers recommendations based on inter-users comparison. In the work of Choi et al.[5], the users' purchase patterns (e.g. ratings and items) are derived by sequential pattern analysis to collaboratively recommend items to users.

Other than the approaches discussed above, a number of hybrid approaches to personalization have also been proposed. Hybrid recommender systems [4] combine content-based and collaborative filtering features in order to overcome the limitation of both techniques. Liu et al.[12] combine user-based and item-based methods to build a hybrid recommendation of movies in P2P networks. When the content of the description is not obvious, a collaborative approach increases the system's precision

2.2 Trust and Reputation

There has been an increasing interest of researchers in the area of trust-based recommendations. For instances:

Golbeck et al [8] applies the concept of trust to web-based social networks. They describe how trust can be computed and how it can be used in applications. Specifically, they proposed two algorithms for inferring trust relationship between individual who are not directly connected to the network. The researchers apply their technique to the TrustMail application, which is an email client that uses the trust algorithm to sort email message in the user's inbox depending on the ratings in the trust network.

Massa et al [13] introduce a trust-aware recommendation architecture which relies on users explicitly stating trust values for other users. They also user a trust propagation technique applied to the network of users, to estimate a trust weight that can be used in place of the similarity weight. An evaluation of their system on the Epinion.com dataset reveals that their system it successful in lowering the mean error on predictive accuracy for cold start users. The authors also define a trade-off situation between recommendation coverage and accuracy in the system.

Various definitions have been presented for trust and reputation. For examples:

Josang et al [9] have defined trust as "the extent to which one party is willing to depend on something or somebody in a given situation, even though negative consequences are possible."

Mui et al [15] defines trust as "a subjective expectation an agent has about another's future behavior based on history of their encounters."

Donovan et al [16] has defined trust as "the reliability of a user to deliver accurate recommendation the past". If a user has a history of making accurate recommendation, they can be viewed as more trustworthiness than a user who made poor recommendation. A user's behavior will determine their trustworthiness.

Reputation for a person is defined as "what is generally said or believed about a person's character or standing.[9]" and they also defined Reputation can be considered as the collective measurement of trustworthiness based on referral or ratings of the members in the community. The basic idea of the trust and reputation system is to derive a score for users. Based on these scores, user can decide whether or not to interact with a user. The main difference between trust and reputation is that trust reflects a subjective view of a user's trustworthiness, whereas reputation reflects the view of whole community.

In this paper, rather than have the user explicitly state trust values, the recommender system will implicitly calculate trust between users. User may not want to expend effort on assigning trust values to other users, but they should still receive the benefit of quality recommendation. The proposed trust metric in this work is a more refined measurement of determining the accuracy of a recommendation. Furthermore, the proposed system also incorporates a reputation value when choosing recommendation partners.

3 Proposed System Model

3.1 Similarity Measurement

This strategy simulates the traditional collaborative filtering approach and will be used as a baseline. From the historical rating data, similarity values are calculated between each pair of users, Eq.(1). The neighborhood is formed by identifying the top-k most similar users. Finally, rating predictions are generated using the Resnick formula, Eq. (2).

Measuring similarity or correlation between users plays a core task in the traditional collaborative filtering recommender system. The first step in the recommendation process is to compute similarity values between each pair of users. The most widely used formula to compute similarity is the Pearson correlation [3].

$$Sim(c,p) = \frac{\sum_{i\in I}(r_{c,i}-\bar{r_c})(r_{p,i}-\bar{r_p})}{\sqrt{\sum_{i\in I}(r_c-\bar{r_c})^2\sum_{i\in I}(r_{p,i}-\bar{r_p})^2}} \tag{1}$$

Let $Sim(c, p)$ represent the similarity between users c and p, where I is the set of items rated by both users, $r_{c,i}$ is the rating user c gave to item i, $r_{p,i}$ is the rating user p gave to item i, $\bar{r_c}$ is the average rating of user c, and $\bar{r_p}$ is the average rating of user p. The results obtained from the Pearson correlation formula range from -1 for negative correlation to +1 for positive correlation. This formula measures the extent to which there is a linear relationship between two variables [6]. For the purposes of collaborative filtering, we look at all co-rated items between user c and user p. If two users have rated many items in common, then they will have a high similarity value. If two users have not rated any item in common, then a similarity value cannot be computed.

Once similarity values have been calculated between every pair of users, we want to identify all users who are the most similar to the target user. The idea is that the most similar users will also be the best recommendation partners. This technique is referred to as, k-Nearest-Neighbors (kNN) [1]. Only the top-kmost similar users will be added to the target user's neighborhood. The neighborhood for the target user is responsible for making recommendations. This network of users will remain static until additional ratings are added to the database, at which point the similarity values should be recalculated and the users in the network could potentially change.

Remember that the goal of collaborative filtering is to predict the ratings of the target user. If the system is able to accurately predict what the target user will rate an item, then the system can recommend desirable items to the target user. The final step in the collaborative filtering process is to generate the rating prediction for the target user on a specific item, i. Once the neighborhood for the target user has been formed, we aggregate the rating information of each neighbor to generate the prediction value. Resnick et al.[17] have proposed a widely used method for computing a prediction value, which has been commonly referred to as the Resnick formula:

$$P_{(c,i)} = \bar{r_c} + \frac{\sum_{p \in M} sim(c, p)(r_{p,i} - \bar{r_p})}{\sum_{p \in M} |sim(c, p)|} \tag{2}$$

Let $P_{c,i}$ represents the predicted rating, where M is the set of all users who belong to target user's neighborhood and have rated item i. Target user, c, for whom predictions are being computed, does not belong to M. $sim(c, p)$ is defined by Eq.(1), $r_{p,i}$ is the rating user p gave to item i, $\bar{r_p}$ is the average rating of user p, and $\bar{r_c}$ is the average rating of user c. In the Resnick formula, each neighbor contributes their rating of item i, and each contribution is weighted according to the specific degree of similarity the neighbor shares with the target user[10]. It is important to note that a neighbor must have rated item i in order to participate in the rating prediction. Thus, if none of the neighbors have rated item i, then a rating prediction cannot be computed. This is a major limitation to traditional collaborative filtering, which will be discussed in the next section. The rating predictions determine which items are recommended to the target user; therefore the quality of recommendations relies on the accuracy of the predictions. In the normal operation of a recommender system, a set of rating predictions are sorted and the items with the highest values are used for recommendation purposes. To measure the accuracy of the recommendations, the target user must submit an item rating, which can then be compared to the predicted rating.

3.2 Trust Metric

For this research, trust has been defined as the ability of a user to provide accurate recommendations. Trust values will be calculated between each pair of users and the trust values are asymmetric. For instance, Alice may view Bob as very trustworthy and Bob may view Alice as untrustworthy. The trust values which a target user holds for all other users will vary over time; this represents the personalization of trust. Over the normal operation of the recommender system, many items will be recommended to a target user, but the target user will only provide ratings to some of the recommended items. Once a rating is received from the target user, we can determine the accuracy of the recommendations by comparing the predicted rating to the user's actual rating. Through the process of comparing predicted and actual ratings, the trust values between users are updated accordingly. As discussed in Sec. 3.1, a rating prediction for a target user is generated using the target user's neighborhood and applying the Resnick formula, Eq. (2). This means that all users in the neighborhood will be contributing to the rating prediction. However, the trust values are supposed to represent the trust that the target user holds for a specific user, p. Therefore, when calculating the trust value, we are only interested in the accuracy of user p's contribution to the overall rating prediction. To capture user p's contribution, the Resnick formula is modified to generate a rating prediction where user p is the sole contributor in Eq 3:

$$P_{c,i} = (\bar{r_c} - \bar{r_p}) + r_{p,i} \tag{3}$$

Let $P_{c,i}$ represents the rating prediction generated from user p, on item i, for target user, c. Where, $\bar{r_c}$ is the average rating of user c, $\bar{r_p}$ is the average rating of user p, and $r_{p,i}$ is the rating user p gave to item i. Once user $p's$ prediction has been generated, the item-level trust, Eq. (4), measures the accuracy of the predicted rating in comparison to the actual rating:

$$T_c(p,i) = 1 - \frac{|P_{c,i} - r_{c,i}|}{r_{max} - r_{min}} \tag{4}$$

Let $T_c(p,i)$represents the item-level trust value, where p is the user who provided the rating prediction on item i, and c is the target user. $P_{c,i}$ is the predicted rating made by user p, $r_{c,i}$ is the target users actual rating, r_{max} is the top of the rating scale, and r_{min} is the bottom of the rating scale. In a collaborative filtering environment which uses a 5-star rating scale, r_{max} would be 5 and r_{min} would be 1. The results for the item-level trust range from 0 to 1, where a larger value means the prediction was more accurate. Remember that trust is defined as the ability of a user to provide accurate recommendations. The item-level trust measures accuracy of user $p's$ prediction for a particular item. An overall trust value for user c and p is calculated by taking the cumulative average of all item-level trust scores in Eq (5):

$$T_c(p) = \frac{\sum_{i \in I} T_c(p,i)}{M} \tag{5}$$

Let $T_c(p)$ represents the overall trust which user c holds for user p, where I is the set of items that user p recommended to user c, $T_c(p,i)$ is the item-level trust for each item i, and M is the total number of items in set I. The results for the overall trust value range from 0 to 1, where a high value means user p has provided very accurate recommendations in the past. The overall trust score indicates the accuracy of user $p's$ recommendations in the past; it also indicates whether or not the target user should trust user p in the future. For this reason, the trust metric provide valuable information when choosing recommendation partners. Additionally, the overall trust value will be updated each time the target user provides a rating to a recommended item. The more ratings that the target user provides, means the trust values become more reliable.

3.3 Reputation Metric

Reputation is measured by how the entire community views an individual. Every user will have a reputation value, which is calculated as the harmonic mean between the average trust score and the experience of the user. To calculate the reputation value for user p, we need to average the trust values that every member in the community holds for user p in Eq (6):

$$T_{avg}(p) = \frac{\sum_{c \in N} T_c(p)}{|N|} \tag{6}$$

Let $T_{avg}(p)$ represent the average trust value that the community has for user p, where N is all set of all users in the community, and $T_c(p)$ is the overall trust value that user c holds for user p. The trust value between user c and p represents the accuracy of $p's$ recommendations to user c. Therefore, the average trust value will indicate $p's$ recommendation accuracy across the entire community. It is possible for user p to have a very high average trust value, even if the total number of recommendations is very low. For instance, Alice and Bob could have identical average trust values of 0.95, yet Alice has contributed 200 recommendations and Bob has only contributed 10 recommendations. Even though their average trust values are the same, Alice would be a more reputable user in the community. For this reason, we need to take into account the number of recommendations that user p has contributed to the community; this will be calculated as Experience in Eq 7:

$$Exp(p) = \frac{N_p}{N_{max}} \tag{7}$$

Let $Exp(p)$ represents the experience of user p, where N_p is the total number of recommendations that user p has contributed, and N_{max} is the maximum number of recommendations contributed by a user. The results for experience will range from 0 to 1,where a larger value means the user has more experience. Once the average trust score and experience have been calculated, the reputation value for user p can be computed using Eq 8:

$$rep(p) = \frac{2(Exp(p))(T_{avg}(p))}{Exp(p) + T_{avg}(p)} \tag{8}$$

Let $rep(p)$ represents the reputation value for user p, where $Exp(p)$ is user $p's$ experience, and $T_{avg}(p)$ is the average trust for user p. The advantage to using the harmonic mean is that it is robust to large differences between the inputs [16], so a high value will only be calculated if both the average trust and experience values are high. The reputation values will be visible to the entire community. Reputation will be particularly valuable for new users, who have not provided any ratings. Without a rating history the recommender system cannot generate similarity values or trust values. In this case, the recommender system could utilize the most reputable users for generating predictions and recommending items.

The proposed strategy requires that trust, reputation, and similarity values are available for all users. The neighborhood for the target user c, is formed by identifying the community members who have the highest combined value of trust, reputation, and similarity. By performing a simple summation of the three values, it gives an overall view into the past history of each user. If a user has a high trust value, high reputation value, and low similarity value, we need to take into account all three values in order to choose the best recommendation partners.

Once the neighborhood is formed, the rating prediction is generated using a modification of the Resnick formula. First, the similarity value, trust value and reputation value are combined to produce a compound weighting using Eq. (9). This combined simiarity/trust/reputation values, and replaces the traditional formula in the Resnick formula, all other variables remain the same, in Eq. (10). The overall process of the proposed method is shown in figure 1.

$$W(c,p) = \frac{3(Sim(c,p)(T_c(p))(rep(p))}{Sim(c,p) + T_c(p) + rep(p)} \tag{9}$$

$$P_{c,i} = \bar{r_c} + \frac{\sum_{p \in N} W(c,p)(r_{p,i} - \bar{r_p})}{\sum_{p \in N} |W(c,p)|} \tag{10}$$

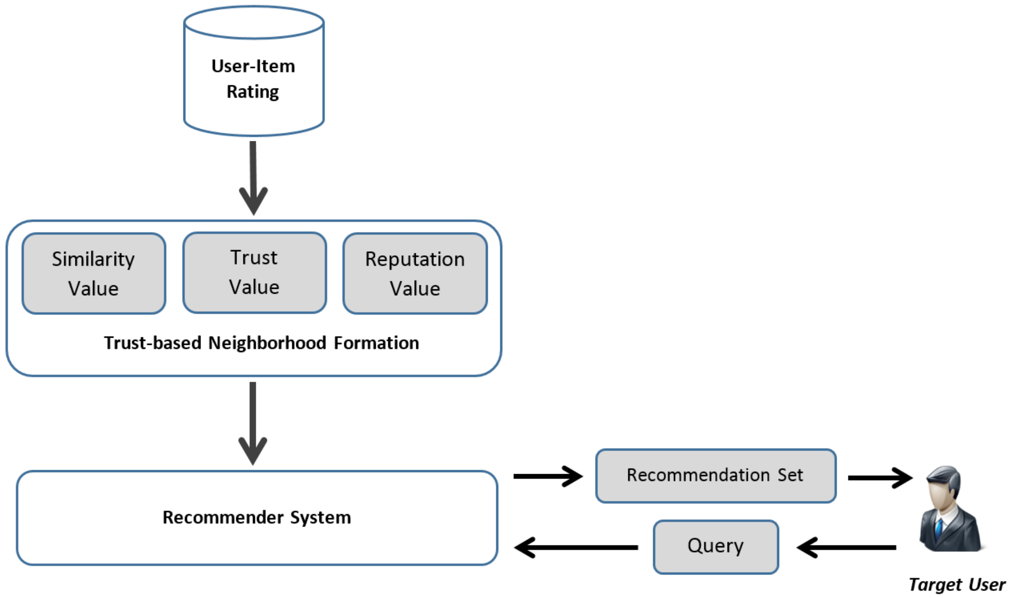

Figure 1: Architecture of proposed model

4 Experiments

The goal of recommendation is to accurately predict what a user will rate for an item, if the recommender system can predict a user's ratings, then it can suggest item which the user will rate highly. This section presents 2 separate experiments conducted for evaluating the performance of the proposed method.

4.1 Data Source

We conducted our experiment using MovieLens dataset [14]. This dataset is the most widely used dataset for collaborative filtering research. The MovieLens dataset was collected by the GroupLens Research Project at the University of Minnesota. The dataset consist of 100,000 ratings, assigned by 943 users on 1682 movies. The users assign numerical ratings in the range from 1 to 5. These numerical ratings can be interpreted as follows: 1 – Bad, 2 – Average, 3 – Good, 4 – Very Good, 5 – Excellent. Each user selected has rated at least 20 movies. with an average number of rating 106.

For each experiment, the MovieLens dataset is divided into a training set and test set. The ratings in the training set are used as explicit ratings from the user, which are contained as part of the historical rating database. The ratings in the test set are treated as unseen ratings that the system would attempt to predict. By comparing the predicted rating with the actual rating we can measure the prediction accuracy of each recommendation strategy.

Before evaluating the accuracy of our proposed trust-based recommender system, we need to build up the trust values, reputation values, and similarity values. Usually the trust values would be built on-the-fly during the normal operation of the recommender system, but for our experiments we need to construct these values in advance using only the training data. The trust values are calculated by running a $leave-one-out$ [16] training session. In the training dataset, we hide one rating at a time and then have each user make an individual rating prediction. By comparing the hidden rating with the predicted rating we can calculate a trust score between the two users.

Once the trust values have been constructed,we can calculate the reputation values, as described in

Sec(3.2) . The similarity values are calculated using Pearson's correlation formula.

4.2 Metrics

The effectiveness of recommender system is measured by the accuracy of prediction it makes. We will use the Mean Absolute Square (MAE). MAE measures the average absolute deviation between a predicted rating and the user's true rating [6]. A small value of MAE implies high prediction accuracy. MAE is a widely used and accepted metric for evaluating the performance of recommender systems. In Eq. (11), P_i represents the predicted rating, r_i represents the user's actual rating, and N represents the total number of items for which the recommender system made a prediction. MAE is a very simple, yet effective way to measure the prediction accuracy of the recommender system. It has been studied and reported that every 1 one-hundredth (0.01) reduction in the MAE, provides a significant improvement in the quality of the recommendations [6].

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{11}$$

4.3 Dataset View

When evaluating the performance of recommender systems, the dataset is an important consideration. Collaborative filtering recommender systems incorporate learning algorithms which operate on statistical data. The performance will vary based on the amount of learning data available. As the quantity of learning data increases, the quality of the predictions should increase as well. Different recommendation strategies will reach an acceptable MAE value at different rates. Some strategies may only need a small amount of data to make decent predictions, while other strategies may need a large amount of data.

In addition to the amount of data, the type of user is also significant. Several researchers have shown that an algorithm will perform differently for different types of users. For example, in a real-world application there are users who will use the application very frequently and there are users who will use the application very rarely. Our experiments conducted in 2 separate scenarios;

In the first scenario, we will test the recommendation strategies on five different types of input data. Each type of input data will provide a different view and insight into how the recommender system performs. First, we will look at new users, who are users that provided only 5 ratings. For this scenario, the recommender system will be making rating predictions for a target user, based on very little knowledge about that user. Second, we will look at heavy users, who are users that provided 50 or more ratings. For the remaining dataset views, we will randomly split the data into the training set and test set, with each view having a different percentage of training data. The three views will be split as follows: 80% training/20% test data; 50% training/50% test data and 10% training/ 90% test data. These dataset views will give an indication for the learning rate of the recommendation strategies. To validate our results, we will perform each experiment five times. Therefore, five distinct dataset are created for each of the five dataset views: "new user" dataset, "heavy user" dataset, 10/90 data split, 50/50 data split, and 80/20 data split.

And in the second scenario, we will perform the result comparison of MAE between our proposed method with Trust-based Collaborative Filtering [10]. In that paper, they present a variation of kNN, the trusted k-nearest recommenders (or kNR) algorithm, which allows users to learn who and how much to trust one another by evaluating the utility of the rating information they receive. In order to do the similar experiment with them, we set the same parameters on MovieLens 80/20 data split.

Types of input data	Traditional CF	Trust Method	Proposed Method
80/20 Test 1	0.9076	0.8507	0.7446
80/20 Test 2	0.9025	0.8467	0.7428
80/20 Test 3	0.9061	0.8420	0.7493
80/20 Test 4	0.9050	0.8453	0.7409
80/20 Test 5	0.8996	0.8464	0.7392
Average	0.9041	0.8462	0.7433

Table 1: Average MAE values for the 80% training, 20% test, dataset view

Types of input data	Traditional CF	Trust Method	Proposed Method
New Users	0.9701	0.8664	0.8315
Heavy Users	0.9316	0.8648	0.7218
10/90 data split	0.9429	0.9368	0.7751
50/50 data split	0.9417	0.8853	0.7546
80/20 data split	0.9041	0.8462	0.7433
Average	0.9380	0.8799	0.7652

Table 2: Average MAE values for different recommendation strategies on different dataset views.

4.4 Results

The MAE results from the 80/20 data split are shown in Table (1) for the first scenario. The average MAE was calculated from the five experiment runs. As you can see, the MAE gradually improved as the recommendation strategy became more complex. The proposed method performed the best with an average MAE of 0.7433. It has a more significant impact when there is less data available.

Instead of showing the result of the five experiments runs for each of the five dataset views, we consolidate the MAE for each dataset view into Table (2):

By looking at each individual recommendation strategy, we can see how the performance of a recommendation algorithm depends on the input data. The results from the 10/90, 50/50, and 80/20 data splits, indicate how fast the algorithms "learn" and begin to generate more accurate predictions. With only 10% training data the Similarity strategy performs poorly with an MAE of 0.9429. By increasing the training data to 80% the similarity-based algorithm provides an average error of 0.9041. This is an expected result, since one of the known drawbacks of using similarity values is that sparse data which means unreliable neighborhood formation that can lead the recommender system to be unable to find highly similar users for the target user, thus the rating prediction is less accurate.

The new user and heavy user dataset views provide insight into the performance of each recommendation strategy on a specific type of user. For the heavy user dataset we see the same trend as the other dataset views. As the recommendation strategy increases in complexity, the MAE value decreases which represents an improvement in prediction accuracy.

The new user dataset shows some interesting results. First, except the traditional CF, we see a very minimal difference in the MAE value across each of the recommendation strategies. The MAE values range from 0.9701 to 0.8315. These results show that it is very difficult to make accurate recommendations for a user who has provided little information. Utilizing the trust values does provide an improvement over the similarity-based algorithm. And with the inclusion of trust values and reputation values, the proposed strategy provides an improvement in prediction accuracy. When there is little knowledge

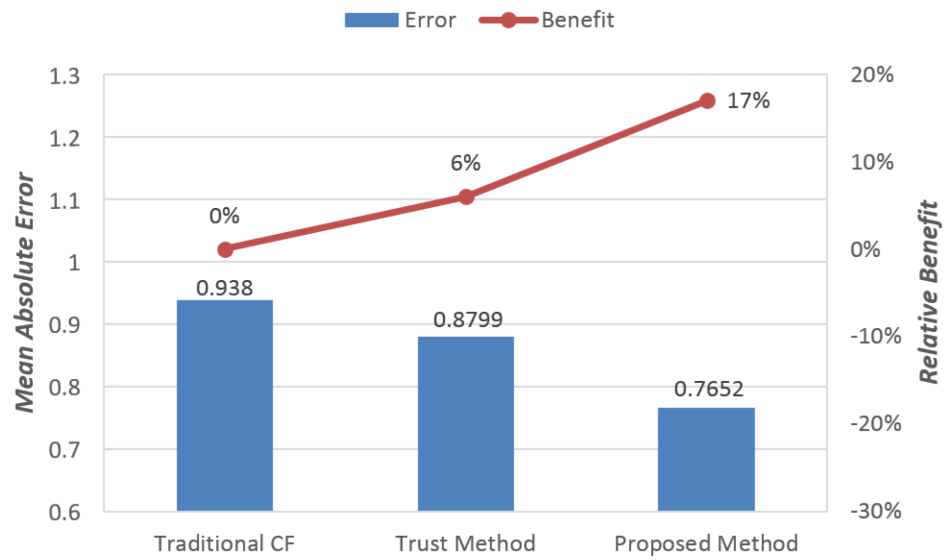

Figure 2: The average prediction error and relative benefit compared to Traditional CF

about the user, it is the reputation which provides invaluable information about the community. A recommender system cannot always generate a similarity-network or trust-network, but the recommender system can always generate a network of reputable community members. Remember that similarity and trust values are calculated between pairs of users, but reputation values are globally available for the entire community to see.

Another interesting result from the new user dataset is the performance of the Similarity strategy. The Similarity strategy provided its least accurate predictions on the new user dataset. System should perform better as the amount of data increases. The heavy users have provided 50 or more ratings, and therefore should have more reliable similarity values with other community members. The new users have provided 5 ratings and the similarity values would be less reliable. Since the Similarity strategy performed worse for the new users and the heavy users, it reinforces the fact that similarity is not a good measurement for choosing recommendation partners. The MAE for the new user dataset and the heavy user dataset was and 0.9701 and 0.9316 respectively.

To give a visual indication of the performance of the recommendation strategies, $Figure(2)$ shows the average MAE for each recommendation strategy, and the relative benefit compared the traditional collaborative filtering strategy. As the recommendation algorithms increase in complexity, the relative benefit increases as well. The Trust strategy provides 6% improvement, the proposed strategy provides 17% improvement in prediction accuracy.

On the other hand, for the second scenario, Accordingly to our MAE results conducted on MovieLens dataset for 80/20 data split (80% - training, 20% - test data) comparing to their result Trust-based Collaborative Filtering in Table (3), we can see that our proposed method outperform better on every single MAE value when the neighbor is increasing.

And in $Figure(3)$, which is represented as a graph derived from Table 3, it outperforms than the Trust-base Collaborative Filtering which less data sparseness. This means that our proposed method will provide and suggest a better predication of recommendation list of items for users.

No of Neighbor(s)	Trust-based CF	Proposed Method
1	0.8705	0.7538
2	0.8226	0.7461
5	0.7913	0.7419
10	0.7821	0.7410
30	0.7791	0.7449
50	0.7794	0.7499
100	0.7804	0.7531

Table 3: Influence of various size of nearest neighbor set on predictive validity

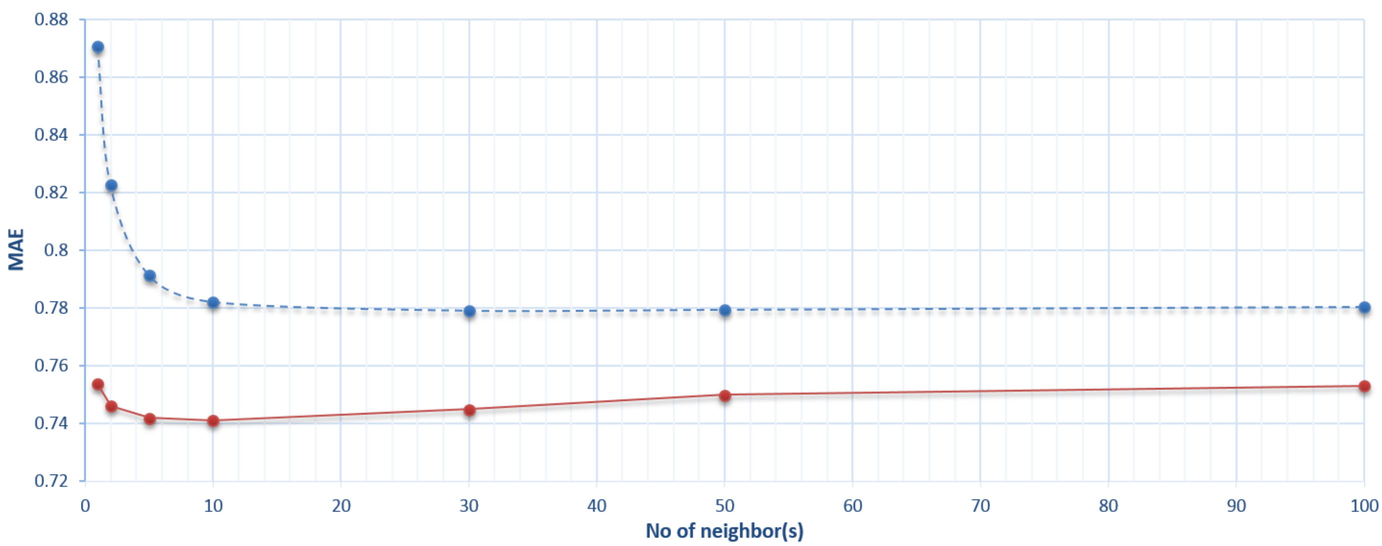

Figure 3: MAE on each algorithm (A small value, a better performance)

5 Conclusion

Trust is the concept that is starting to attract the increasing attention by many researchers and is being implemented in many online community systems.

This paper describes an improved method for recommender systems. We have shown that by incorporating Similarity, Trust and Reputation into the recommender systems, it can lead to a promising accuracy and quality of recommendations in an effective way. The results from various simulations using MovieLens dataset show that the proposed method have 16% reduction in error compared to the traditional collaborative filtering technique and performs better than the Trust-based collaborative filtering in other research. Throughout the experiment analysis, we found that the data sparseness has less impact on the proposed system, which helps to increase the accuracy and quality of recommendations.

6 Future Works

There are some directions for the future studies. First, the proposed model can be advanced by exploiting other techniques, such as artificial intelligence and machine learning. Second, the targets of recommendations can be further extended, such as social networking-driven recommendations. Finally, the proposed model should be further validated by experimenting with more datasets.

7 Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No.201302970).

References

[1] E. Aimeur and F. Onana. Better control on recommender systems. In *Proc. of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC-EEE'06), San Francisco, California, USA*, pages 38–38. IEEE, June 2006.

[2] S. Anand and B. Mobasher. Intelligent techniques for web personalization. In *Proc. of the Intelligent Techniques for Web Personalization In Intelligent Techniques for Web Personalization*, volume 3169, pages 1–36. Springer Berlin Heidelberg, 2005.

[3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98), San Francisco, California, USA*, volume 2782, pages 43–52. Morgan Kaufmann Publishers Inc, 1998.

[4] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[5] K. Choi, D. Yoo, G. Kim, and Y. Suh. Hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, 11(4):309–317, July 2012.

[6] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, January 2004.

[7] Z. Huang, W. Chung, and H. Chen. A graph model for E-commerce recommender systems. *Journal of the American Society for Information Science and Technology*, 55(3):259–274, February 2004.

[8] J. H. Jennifer Golbeck. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *Proc. of the 14th International Conference on Engineering Knowledge in the Age of the Semantic Web (EKAW'04), Whittlebury Hall, UK, LNCS*, volume 3257, pages 116–131. Springer-Verlag, October 2004.

[9] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, March 2007.

[10] N. Lathia, S. Hailes, and L. Capra. Trust-based collaborative filtering. In *Proc. of the The International Federation for Information Processing in Trust Management (IFIPTM'08), Trondheim, Norway, IFIP*, volume 263, pages 119–134. Springer US, June 2008.

[11] Y.-H. Lee, P. J.-H. Hu, T.-H. Cheng, and Y.-F. Hsieh. A cost-sensitive technique for positive-example learning supporting content-based product recommendations in B-to-C e-commerce. *Decision Support Systems*, 53(1):245–256, April 2012.

[12] Z. Liu, W. Qu, H. Li, and C. Xie. A hybrid collaborative filtering recommendation mechanism for P2P networks. *Future Generation Computer Systems*, 26(8):1409–1417, October 2010.

[13] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proc. of the 2007 ACM Conference on Recommender Systems(RecSys'07), Minneapolis, Minnesota, USA*, pages 17–24. ACM, October 2007.

[14] MOVIELENS-DATA. Movielens dataset. `http://www.grouplens.org/data/`.

[15] L. Mui and M. Mohtashemi. A computational model of trust and reputation. In *Proc. of the 35th Hawaii International Conference on System Science (HICSS'02), Big Island, Hawaii, USA*, pages 2431–2439. IEEE, January 2002.

[16] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proc. of the 10th International Conference on Intelligent User Interfaces (IUI'05), San Diego, California, USA*, pages 167–174. ACM, January 2005.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94), Chapel Hill, North Carolina, USA*, pages 175–186. ACM, October 1994.

[18] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997.

[19] G. Rowe and G. Wright. Expert opinions in forecasting: The role of the delphi technique. In *Principles of Forecasting-A Handbook for Researchers and Practitioners, International Series in Operations Research & Management Science*, volume 30, pages 125–144. Springer US, 2001.

[20] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(4):4:2–4:2, January 2009.

Author Biography

ChanChamnab Than obtained the B.S. degree in the Department of Computer Science at National Polytechnic Institute of Cambodia. Now he is a master degree student in the School of Computer Science and Engineering at ChungAng University, Seoul, Korea. His research interests include E-commerce and Social Network.

Sangyong Han is a professor of the School of Computer Science and Engineering, ChungAng University, Seoul, Korea. He received Bachelor of Engineering in College of Engineering from Seoul National University in 1975, and the Ph.D. degree in 1984. From 1984 to 1995, he worked at Poughkeepsie Lab. and Watson Research Center in IBM, USA. His research interests include next generation web, information retrieval, and optimization.