A recommender system based on trust and semantics in collaborative systems using a new measure of association

Seyedeh Homa Alizadeh
Computer Engineering Department
Imam Reza International University
Mashhad, Iran
homa.alizadeh@hotmail.com

Majid Vafaei Jahan
Computer Engineering Department
Islamic Azad University
Mashhad, Iran
vafaeijahan@mshdiau.ac.ir

N. R. Arghami
R&D Department
FAMA Technology
Naperville, IL, USA
arghami@yahoo.com

Abstract—One of the most popular techniques used in recommender systems is collaborating filtering. In this technique it is usual that Pearson’s correlation is used to find the similarity between users. It is a known fact that Pearson’s correlation is not suitable for measuring the strength of non-linear relations. Since Spearman’s correlation is in fact Pearson’s correlation applied to ranks and does not work well in non-monotone relationships and since measures like Kendall’s tau do not work well in small samples, we introduce a new measure of association to be used in collaborative systems which we shall call alpha.

Our investigations show it leads to better MAE. We also propose a method by combining Alpha and trust propagation and add a new algorithm to semantic similarity for confronting the problems with cold start and it leads to better coverage.

Keywords: recommender systems, alpha measure of association, semantic similarity, collaborative filtering, cold start

I. INTRODUCTION

Recommender systems help the user by predicting the items that he or she is more interested in and thus save him or her time and effort.

In recommender systems, there are two main concepts, one being user, that is the person who benefits from the service and the other being item, which is to be recommended. The users to whom some items are to be recommended are called active users.

One of the frequently used recommender systems uses collaborating filtering (CF) method. In this method users rate the items usually on a discrete numerical scale from 1 to 5. Here 1 and 5 may indicate the least and the most interest, respectively. In order to provide recommendations, ratings are used to compute the similarities between users and/or items [1,2]. It is usual to use Pearson’s correlation (PC) as the measure of similarity. Since the ratings of two users on the same set of items (or the ratings of two items given by the same set of users) cannot be assumed to have a linear relationship, PC may not be the best choice as a measure of similarity for the problem at hand [3,4,5,6]. Spearman’s correlation is suitable only for monotone relationships and measures like Kendall’s tau do not perform well in small samples [6]. Thus we introduce a new measure of similarity which we call alpha and denote it by α. We shall investigate its properties and in particular we shall see that it has a better MAE.

To increase the efficiency of CF, especially in relation to cold start problem, we use a concept called trust. To this end, users indicate explicitly their degree of trust in other users, or this can be evaluated by the system implicitly [7,8,9].

Another way to compute the similarity between items is the semantic-based technique. Semantically similar items to the user’s items of interest are then recommended. For this technique ontology, or item taxonomy is used. This technique helps us deal better with the cold start problem and gives better coverage [9,10]. We shall use semantic similarity between users and between items, by using taxonomy, in order to increase coverage and the efficiency of the system.

II. BACKGROUND AND RELATED WORK

A. Content-based recommender systems

Content-based recommender systems make their recommendations based on attributes and descriptions of items and user’s interests profile. Description of items can be structured, semi-structured or non-structured. Structured descriptions are like a table, in which the columns contain the attributes and descriptions of items [11,12]. User’s profile can be provided by the user or can be obtained from user’s feedback on used items [11,13].

B. CF recommender systems

CF is the most popular technique in the current recommender systems. It divides into two methods of user-
based and item-based. Similarities between users are computed from common ratings of items between two users and similarities between items are computed from the ratings given by common users to a pair of items [1,2,9,11].

C. Trust-based recommender systems

Recommender systems based on trust use social networks, in which users rate each other according to the degree to which they trust every other user [8]. This degree of trust can be explicit or implicit. For explicit trust the degree of trust is obtained from the social links between the users and implicit trust is extracted from item ratings [9,14,15,16].

D. Semantic-based recommender systems

Semantic-based recommender systems use semantic attributes of users for generating recommendations. Semantic information, for example, can include item properties, relation between items and metat data. Taxonomy and ontology are two important sources of semantic information. They discover and classify new information about items and user profiles [9,10].

III. THE PROPOSED APPROACH

Fig. 1 shows the structure of the proposed approach. This approach has three modules. Modules are parallel and independent. The first and third modules are user-dependent, while the second module is item-dependent. The inputs are a raw user-item rating matrix, item taxonomy and user taxonomy.

A. User-based similarity and trust module

This module makes recommendations based on similarity and trust of the users. The output of this module is user-item matrix and contains predicted ratings based on this module.

1) Computing similarity and implicit trust of the users

In this section we introduce our new similarity measure alpha and use it to compute similarity and implicit trust of users. Our proposed similarity measure alpha is defined as

\[
\alpha_{ab} = \frac{4n_{ab} - \sum_{j=1}^{n_{ab}} [(r_{aj} - \bar{r}_a) - (r_{bj} - \bar{r}_b)]}{4n_{ab}},
\]

where \( r_{aj} \) and \( r_{bj} \) are ratings given to item \( j \), respectively by users \( a \) and \( b \). \( \bar{r}_a \) and \( \bar{r}_b \) are the mean ratings given by \( a \) and \( b \), respectively, to all items that are rated by each user separately and \( n_{ab} \) is the number of items that have been rated by both \( a \) and \( b \). It is easy to show that \( \alpha \) takes values in the interval \([0,1]\). Values of \( \alpha \) closer to 1 indicate more similarity. We shall use a threshold value of 0.5 for \( \alpha \). The threshold 0.5 is the boundary between dissimilarity and similarity. We have also reached this threshold by experimenting on the dataset under study.

Since when \( r_{aj} = r_{bj} \), for all \( j \), \( \alpha = 1 \), regardless on the number of common items, \( n_{ab} \), we adjust the value of \( \alpha \) by subjecting it to Jaccard formula as in [17], that is we multiply \( \alpha \) by

\[
\text{UJaccard}_{ab} = \frac{n_{ab}}{n_a + n_b - n_{ab}},
\]

where \( n_a \) and \( n_b \) are the total number of items rated by users \( a \) and \( b \), respectively. The final result is

\[
e_{\alpha_{a,b}} = \alpha_{a,b} \times U\text{Jaccard}_{a,b},
\]

where \( \alpha_{a,b} \in [0,1] \).

1) Propagation of trust

When there is not a direct relation between two users, we can use trust propagation technique [7,8,9,14,15,16]. Suppose, for example, that user a trusts user b who trusts user c. Using the trust propagation technique, we can find out how much a trusts c. The formula for trust propagation is

\[
P_{\alpha_{a,c}} = \frac{\sum_{\text{beadj}(a)} e_{\alpha_{a,b}} \times (e_{\alpha_{b,c}} \times \beta_d)}{\sum_{\text{beadj}(a)} e_{\alpha_{a,b}}}, \quad e\alpha \geq \lambda,
\]

where \( \beta_d = (\text{MPDist} - d + 1)/\text{MPDist} \), \( d \in [2,\text{MPDist}] \).

In(4), there are users denoted by b who are trusted by a and c is a user who is trusted by users b. Parameter \( \lambda \) is a tunable parameter. Trust propagates from a to c through b if the degree that a trusts b is equal to or greater than \( \lambda \), where \( \lambda \in [0,1] \).

The degree of trust that is computed from (4) decreases for each additional level of propagation [8]. In(5), \( \beta_d \in [0,1] \).

MPDist is the maximum distance of the user in the beginning point of propagation to the user who receives propagation. Parameter \( d \) indicates the number of users between the user under consideration and the final user.

2) Choosing neighbors

In this section we select the most similar neighbors from the set \( N^{a\alpha} \). To do this we use Top-n method [9,18].

3) Computing prediction weights

We use the following formula to predict the rating of active user \( a \in U \) on the item \( x \in I \).

\[
R^{a}_{a,x} = \frac{\sum_{b=1}^{N_{\text{pea}}} P_{\alpha_{a,b}} \times (r_{b,x} - \bar{r}_b + \bar{r}_a)}{\sum_{b=1}^{N_{\text{pea}}} P_{\alpha_{a,b}}},
\]

where \( N_{\text{pea}} \) is the number of peers of active user \( a \). In(6), \( \bar{r}_a \) and \( \bar{r}_b \) are the mean of ratings given by active users \( a \) and \( b \), respectively. \( r_{b,x} \) is the rating of user \( b \) on item \( x \). The degree of similarity and also of trust of active user \( a \) to its neighbor user \( b \) is denoted by \( P_{\alpha_{a,b}} \in [0,1] \). The set of chosen neighbors for active user \( a \) is denoted by \( N^{a\alpha} \) and \( R_{a,x}^{a} \in [0,1] \) is the predicted rating.

B. The module for the item-based CF similarity and semantic similarity

In this module the recommendations are generated by combining item-based CF and item-based semantic filtering approaches. Also in this module the weight of item reputation is defined and used in the prediction process. The module of this section is quite similar to Shamour’s second module [9].
1) **Computing item-based CF similarity**

In this section, in order to compute item-based CF similarity, we use a combination of cosine similarity [19] and item-based Jaccards [17].

\[
I_{AdjCos_{x,y}} = \frac{\sum_{u=1}^{U_{x,y}}(r_{u,x} - \bar{r}_u) \times (r_{u,y} - \bar{r}_u)}{\sqrt{\sum_{u=1}^{U_{x,y}}(r_{u,x} - \bar{r}_u)^2 \times \sum_{u=1}^{U_{x,y}}(r_{u,y} - \bar{r}_u)^2}}. \quad (7)
\]

\[
I_{Jaccard_{x,y}} = \frac{|U_{x,y}|}{|U_x| + |U_y| - |U_{x,y}|}. \quad (8)
\]

In (7), \(r_{u,x}, r_{u,y} (\in [1,5])\) denote the active user u’s rating of items x and y, respectively. \(\bar{r}_u\) is the mean of all ratings given by user u to all items. \(U_{x}, U_{y}\) and \(U_{x,y}\) are the set of all users who have rated item x, item y and both items x and y, respectively. \(|U_{x}|, |U_{y}|\) and \(|U_{x,y}|\) are the number users in \(U_{x}, U_{y}\) and \(U_{x,y}\), respectively. The final formula of this section is

\[
eICF_{x,y} = I_{AdjCos_{x,y}} \times I_{Jaccard_{x,y}}. \quad (9)
\]

Which gives the item-based CF similarity between item x and the neighbor item y which has an output in [-1,1].

2) **Computing item-based semantic similarity**

First, in order to use semantic information, we construct the item taxonomy in a tree structure. To do this we first identify the main categories of items and we assign items to appropriate categories. An item can belong to more than one category. Similarity between items is determined on the basis of their semantic descriptions in the taxonomy tree [9]. The corresponding formula is

\[
SSim_{x,y} = \frac{C_{11}}{C_{01} + C_{10} + C_{11}}, \quad (10)
\]

for each pair x and y \(\in \mathbb{I}\), by which semantic similarity between the target item x and the neighbor item y, \(SSim_{x,y} (\in [0,1])\) is
computed. C11 refers to the number of categories to which x belongs but y does not and C10 refers to the number of categories to which y belongs but x does not.

3) Computing item reputation

First, we compute item reputation using the formula

$$R_y = \left( \frac{|U_y|}{|U|} \right) \times \bar{r}_y,$$

(11)

in which $U_y$ is the number of users who have rated y, $|U|$ is the total number of users and $\bar{r}_y$ is the mean of ratings of item y.

4) Choosing neighbors

In this section we choose two sets of neighbors, using Top-n method. One set is chosen based on item-based CF similarity, $N^{SeCF}$ and the other is based on item-based semantic similarity, $N^{Sem}$.

5) Computing prediction weights

We use the following formula to predict ratings of active user $a \in U$ on item $x \in I$:

$$R_{a,x}^{SeCF} = \bar{r}_x + \sum_{y=1}^{N^{SeCF}} (eICF_{x,y} \times (r_{a,y} - \bar{r}_y) \times R_y) + \sum_{y=1}^{N^{Sem}} (SSim_{x,y} \times (r_{a,y} - \bar{r}_y) \times R_y)$$

$$\sum_{y=1}^{N^{SeCF}} (eICF_{x,y} + R_y) + \sum_{y=1}^{N^{Sem}} (SSim_{x,y} + R_y)$$

(12)

In (12), $R_{a,x}^{SeCF} \in [1,5]$ and $\bar{r}_x$ and $\bar{r}_y (\in [1,5])$ are the mean of ratings of the target item x and the neighbor item y. eICF$_{x,y}$ (E[-1,1]) indicates the item-based CF similarity of the target item x and the neighbor item y. $SSim_{x,y}$ (E[-1,1]) indicates the item-based semantic similarity of the target item x and the neighbor item y. $N^{SeCF}$ and $N^{Sem}$ are two sets of the closest neighbors of the target item x on the basis of item-based CF similarity and item-based semantic similarity, respectively. $r_{a,y} (\in [1,5])$ refers to rating of active user $a$ on neighbor item $y$.

C. Fusion of predictions

In this module ratings are predicted for the items that have not been rated by active user $a$. The ratings obtained in this module are the fusion of the ratings obtained in the previous two modules.

$$FP_{a,x} = \begin{cases} 0 & \text{if } R_{a,x}^{a} = 0 \text{ and } R_{a,x}^{SeCF} = 0 \\ R_{a,x}^{a} & \text{if } R_{a,x}^{a} \neq 0 \text{ and } R_{a,x}^{SeCF} = 0 \\ R_{a,x}^{SeCF} & \text{if } R_{a,x}^{a} = 0 \text{ and } R_{a,x}^{SeCF} \neq 0 \\ 2 \times R_{a,x}^{a} \times R_{a,x}^{SeCF} & \text{if } R_{a,x}^{a} \neq 0 \text{ and } R_{a,x}^{SeCF} \neq 0 \\ \frac{R_{a,x}^{a} + R_{a,x}^{SeCF}}{R_{a,x}^{a}} & \text{if } R_{a,x}^{a} \neq 0 \text{ and } R_{a,x}^{SeCF} = 0 \end{cases}$$

(13)

D. User’s semantic similarity computation module

In this module we construct user’s taxonomy tree to be able to use user’s semantic information. In this section only job related user’s information is used. Construction of user’s taxonomy tree is similar to that of items, with the difference that in user’s taxonomy tree each user has a job. In this tree the first level is root and various jobs are in the next level and in the next level users are placed in the appropriate category according to their jobs.

In order to compute the semantic similarity of users, we compute the means of the ratings given by users of each job, separately.

In case that, using formula (13), an item remains without rating, this module is used for rating it. To estimate the rating of an item which couldn’t predict it in previous section, the mean of estimated rating for that job is used.

$$FFP_{a,x} = \begin{cases} \frac{\mu_{a,Job}}{x} & \text{if } FP_{a,x} = 0 \\ \frac{FP_{a,x}}{x} & \text{if } FP_{a,x} \neq 0 \end{cases}$$

(14)

In (14), $j$ is the job of the active user $a$ and $x$ is the target item. FFP$_{a,x} \in [0,5]$.

IV. EMPIRICAL RESULTS

A. The dataset

The dataset used in this study is MovieLens in [20]. The sparsity of this dataset is 0.937.

B. Evaluation matrices

Mean Absolute Error (MAE) and Coverage are used to evaluate the recommendation methods [2,13,18]. Smaller values of MAE indicate higher accuracy and higher coverage values are more desirable.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |r_i - \bar{r}_i|.$$  

(15)

Equation (15), needs the set of actual/predicted rating pairs for all the n items available in the test set.

$$\text{Coverage} = \frac{I_p}{n}.$$  

(16)

In (16), $I_p$ denotes the set of items for which a prediction can be made and n is the number of available items.

C. Benchmark algorithm

To evaluate the proposed approach, it should be compared with other methods. For this purpose, we use Shambour’s [9] work. The reason for using this method is this work’s similarity to our proposed approach.
We use a similarity measure based on a new measure of association instead of PC in the first module and add a new user semantic module at the end of the approach.

D. Parameter setup

Based on the experiments that we have done on the dataset MovieLens, we have selected the values of the constant parameters in order to have the best results on accuracy and coverage. The settings \( \lambda = 0.2 \) and MPDist = 3 and \( \alpha > 0.5 \) seem to be the best thresholds for the similarity computations.

E. Evaluation of the proposed system

The evaluations were done in three parts: number of neighbors, the impact of the proposed approach on the user cold start problem and the item cold start problem.

1) Number of neighbors

In Fig. 2 and Fig. 3 the MAE and coverage of the proposed approach and those of Shambour’s method are compared for the number of neighbors from 10 to 90.

![Figure 2](image1.png)

**Figure 2.** MAE of Shambour's system and ours for various numbers of neighbors

![Figure 3](image2.png)

**Figure 3.** Coverage of Shambour's system and ours for various numbers of neighbors

As we see in Fig. 2, the MAE of the proposed approach, for small number of neighbors, is smaller than that of Shambour’s, indicating that under this condition our approach has a better output.

Also in Fig. 3, we see that the coverage of the proposed approach is greater than that of Shambour’s, the differences getting smaller as the number of neighbors becomes larger.

2) User’s cold start

In Fig. 4 and Fig. 5, we investigate the proposed approach in confronting user’s cold start problem. To do this we compute the MAE and coverage of the proposed approach for different ratings of users for items to investigate the cold start.

![Figure 4](image3.png)

**Figure 4.** MAE of Shambour's system and ours for various numbers of ratings for CS users

![Figure 5](image4.png)

**Figure 5.** Coverage of Shambour's system and ours for various numbers of ratings for CS users

As we see in the mentioned figures, MAE of the proposed approach is slightly better, while the improvement in coverage is more significant.

3) Item cold start

We investigate the item cold start problem with the proposed approach in Fig. 6 and Fig. 7. The figures show the MAE and coverage of the methods for different number of item ratings, in relation to item cold start.

![Figure 6](image5.png)

**Figure 6.** MAE of Shambour's system and ours for various numbers of ratings for CS items
As we see in these figures, in both cases the proposed approach has a better output.

V. CONCLUSION AND FUTURE WORK

In this paper we have presented a new approach, based on trust and semantics, for personalization of recommendations. A new measure of association, that we have named alpha, is proposed to replace Pearson’s correlation and our evaluations show better efficiency in terms of MAE, based on alpha. We have also improved the coverage by adding a third module based on semantic similarity. In future work we hope to make a more accurate use of user’s semantic similarity.

To evaluate the proposed approach we compared it with Shambour’s work, because of its similarity to our proposed method. In fact this article is based on his work. We saw that our proposed method compares favorably with that of Shambour’s.

REFERENCES


Figure 7: Coverage of Shambour’s system and ours for various numbers of ratings for CS items