Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences

Manos Papagelis[1,2], Dimitris Plexousakis[1,2], and Themistoklis Kutsuras[2]

[1] Institute of Computer Science, Foundation for Research and Technology – Hellas, P.O. Box 1385, GR-71110, Heraklion, Greece
[2] Department of Computer Science, University of Crete, P.O. Box 2208, GR-71409, Heraklion, Greece
{papaggel, dp}@ics.forth.gr, kutsuras@csd.uoc.gr

Abstract. Collaborative Filtering (CF), the prevalent recommendation approach, has been successfully used to identify users that can be characterized as "similar" according to their logged history of prior transactions. However, the applicability of CF is limited due to the *sparsity* problem, which refers to a situation that transactional data are lacking or are insufficient. In an attempt to provide high-quality recommendations even when data are sparse, we propose a method for alleviating sparsity using *trust inferences*. Trust inferences are transitive associations between users in the context of an underlying social network and are valuable sources of additional information that help dealing with the *sparsity* and the *cold-start* problems. A trust computational model has been developed that permits to define the *subjective* notion of trust by applying *confidence* and *uncertainty* properties to network associations. We compare our method with the classic CF that does not consider any transitive associations. Our experimental results indicate that our method of trust inferences significantly improves the quality performance of the classic CF method.

1 Introduction

Recommendation systems [1] have been a popular topic of research ever since the ubiquity of the web made it clear that people of widely varying backgrounds would be able to access and query the same underlying data. Both research and e-commerce applications have extensively adopted variations of recommendation algorithms in order to provide an intelligent mechanism to filter out the excess of information available to their users. *Collaborative filtering* (CF) [2] has almost certainly been the finest technique of choice for recommendation algorithms. CF tries to identify users that have relevant interests and preferences by calculating similarities among user profiles [3]. The idea behind this method is that, it may be of benefit to one's search for information to consult the behavior of other users who share the same or relevant interests and whose opinion can be trusted.

Regardless of its success in many application settings, the CF approach encounters two serious limitations, namely sparsity and scalability [4, 26]. In this paper we focus on the sparsity problem. The sparsity problem occurs when available data are insufficient for identifying similar users (neighbors) and it is a major issue that limits

P. Herrmann et al. (Eds.): iTrust 2005, LNCS 3477, pp. 224 – 239, 2005.

the quality of recommendations and the applicability of CF in general. The main objective of our work is to develop an effective approach that provides high-quality recommendations even when sufficient data are unavailable.

The remainder of the paper is organized as follows: Section 2 elaborates on the sparsity challenge and explains the weaknesses of already proposed methods for dealing with it. Section 3 presents our methodology that is based on trust inferences., while Section 4 presents experimental evaluation of our work.

2 Problem Statement

The numbers of users and items in major e-commerce recommendation systems is very large [5]. Even users that are very active result in rating just a few of the total number of items available in a database and respectively, even very popular items result in having been rated by only a few of the total number of users available in the database. This problem, commonly referred to as the *sparsity problem*, has been identified as one of the main technical limitations of CF and its further development and adoption. Because of sparsity, it is possible that the similarity between two users cannot be defined, rendering CF useless. Even when the evaluation of similarity is possible, it may not be very reliable, because of insufficient information processed. The cold-start problem emphasizes the importance of sparsity problem. *Cold-start* [6] refers to the situation in which an item cannot be recommended unless it has been rated by a substantial number of users. This problem applies to new and obscure items and is particularly detrimental to users with eclectic taste. Likewise, a new user has to rate a sufficient number of items before the recommendation algorithm be able to provide reliable and accurate recommendations.

There are several methods that have been proposed to deal with the sparsity problem. Most of them succeed in providing better recommendations, but fail to introduce a general model for dealing with sparsity. Most popular approaches proposed include dimensionality reduction of the user-item matrix, application of associative retrieval technique in the bipartite graph of items and users, item-based similarity instead of user-based similarity, and content-boosted CF. The dimensionality reduction approach addresses the sparsity problem by removing unrepresentative or insignificant users or items so as to condense the user-item matrix. More advanced techniques to achieve dimensionality reduction have been proposed as well. Examples include statistical techniques such as Principle Component Analysis (PCA) [7] and information retrieval techniques such as Latent Semantic Indexing (LSI) [8, 9, 10]. However, potentially useful information might be lost during this reduction process. Transitive associations of the associative retrieval technique [11], even if they have been successfully employed to deal with the sparsity problem, fail to express the subjective notion of the associations. Item-based [12, 13] in addition to Content-boosted CF [13, 14] approaches require additional information regarding items as well as a metric to compute meaningful similarities among them [25].

Our research work provides an alternative approach to deal with sparsity problem. Instead of reducing the dimension of the user-item matrix, in an attempt to make it more informative, we propose a method that permits to define transitive properties between users in the context of a social network. The consideration of these properties leads to extra information accessible for recommendation purposes. Our approach focuses on developing a computational model that permits the exploration of transitive user similarities based on trust inferences for addressing the sparsity problem.

3 Methodology

3.1 Social Networks in Recommender Systems

CF has been successfully employed to express the “word-of-mouth” paradigm in a computational context [15]. Common interactions that take place in a typical recommendation system include ratings, transactions, feedback data etc. For the rest of the paper we assume without loss of generality that interactions are based on rating activity. Based on these interactions, it is possible to express similarity conditions between pairs of users, according to the subset of their co-rated items. We view these similarity conditions as associations between users. It is then possible to consider these associations as links of a *social network*. If we define as user-item matrix the matrix having as elements the ratings of users to items, then a user’s model [16] is represented in this matrix as an *n*-dimensional vector, where *n* is the number of items in the database. Figure 1 illustrates the process of the network construction, where a user’s rating activity is used to define network associations.

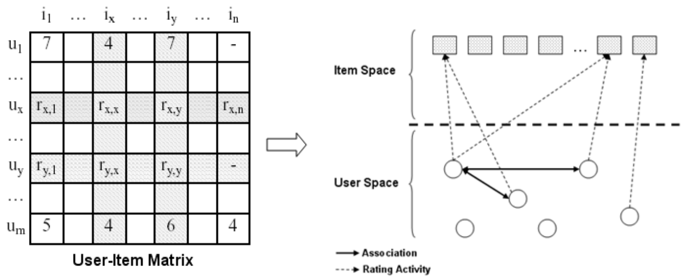

Fig. 1. Underlying Social Networks in Recommender Systems

As theories on social networks find application in completely diverse research areas, we need to properly describe their particularities in our context and most importantly identify the process of *membership* and *evolution*.

Membership: A user joins the underlying social network by submitting at least one rating to an item that has previously been rated by another user.

Evolution: Users’ ratings to items are enabling the construction of new associations between users and thus new links in the underlying network are considered.

3.2 Trust Through User-to-User Similarity

We think of the associations between users as an expression of established *trust* between each other, as far as the specific application area is concerned. Since trust is defined in the context of similarity conditions, the more similar the two users are the greater their established trust would be considered [17]. In order to compute the similarity between users, a variety of similarity measures have been proposed, such as Pearson correlation, cosine vector similarity, Spearman correlation, entropy-based uncertainty and mean-square difference. However, Breese et al in [18] and Herlocker et al. in [19] suggest that Pearson correlation performs better than all the rest.

If we define the subset of items that users u_x and u_y have co-rated as I={i_x: x=1, 2, …, n}, r_{u_x,i_h} as the rating of user u_x to item i_h and $\overline{r_{u_x}}$, $\overline{r_{u_y}}$ as the average ratings of users u_x and u_y respectively, then the established trust between two users is defined as the Pearson correlation [20] of their associated rows in the user-item matrix (Eq. 1).

$$T_{x\to y} = sim(u_x, u_y) = \frac{\sum_{h=1}^{n'} (r_{u_x,i_h} - \overline{r_{u_x}})(r_{u_y,i_h} - \overline{r_{u_y}})}{\sqrt{\sum_{h=1}^{n'} (r_{u_x,i_h} - \overline{r_{u_x}})^2}\sqrt{\sum_{h=1}^{n'} (r_{u_y,i_h} - \overline{r_{u_y}})^2}} \tag{1}$$

3.3 Trust Inferences

Due to the number of ratings that exist in recommendation systems, underlying social networks are very sparse. There are cases in which insufficient or loss of information is detrimental for the recommendation algorithms. Consider, for example, the case in which associations between users are based on very few data or the case in which there aren't any *k* users to employ in a *k*-nearest neighborhood algorithm. A motivating example is illustrated in Figure 2(a). Suppose that users *S*, *N* have rated item I_1 and users *N*, *T* have rated I_2. Classic CF will associate user *S* with user *N* and user *N* with user *T*, but not user *S* with user *T*. However, a more sophisticated approach that incorporates transitive interactions would recognize the associative relationship between user *S* and user *T* and infer this indirect association. To deal with this problem, we adopt a method of inferring trust between users that are not directly associated to each other. Thus, in the example, it is possible to infer trust between the source user *S* and the target user *T* through the intermediate user *N*. According to this process, trust is propagated in the network and associations between users are built, even if they have no co-rated item.

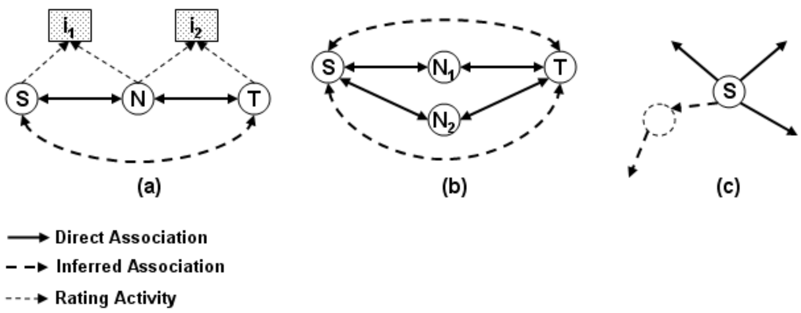

Fig. 2. Trust Inferences

Trust Paths

Propagation of trust [21, 22] implies the existence of *trust paths* in the network. Combination of consecutive direct associations between all intermediate users creates a trust path from a source user to a target user. Trust paths can be of variable *length*, depending on the number of associations that one needs to traverse in order to reach the target user. If *k* associations need to be traversed then the path is considered to be of length *k*. Direct associations are of length *1*, while when the target user is not accessible from the source user, the length of the supposed path is considered infinite.

While computation of trust in direct associations is based on user-to-user similarity, for length-k associations we need to adopt a transitivity rule that facilitates the computation of the inferred trust between the source user and the target user. If we define as N={Ni: i=1, 2, …,k} the set of all intermediate nodes in a trust path that connects user S and user T, then their associated inferred trust is given by Equation 2.

$$T_{S \underset{N_1 \to \ldots \to N_k}{\to} T} = \left(\left(\left(\left(T_{S \to N_1} \oplus T_{N_1 \to N_2} \right) \oplus \ldots \right) \oplus T_{N_{k-1} \to N_k} \right) \oplus T_{N_k \to T} \right) \tag{2}$$

For example, in order to compute to what degree user S trusts user T in the example of Figure 2(a), we need to compute the inferred trust $T_{A \underset{B}{\to} C} = T_{A \to B} \oplus T_{B \to C}$. In Equation 2, we employ the symbol $\oplus$ to denote that we need to apply a special operation in order to compute the inferred trust in the path. If I_x is the set of items that user u_x has rated, and $n(I_x)$ is the cardinality of the set I_x, then Equation 3 interprets the special operation employed.

$$T_{S \underset{N}{\to} T} = T_{S \to N} \oplus T_{N \to T} = \oplus \left(\frac{n(I_S \cap I_N)}{n(I_S \cap I_N) + n(I_N \cap I_T)} \left| T_{S \to N} \right| + \frac{n(I_N \cap I_T)}{n(I_S \cap I_N) + n(I_N \cap I_T)} \left| T_{N \to T} \right| \right)$$

$$where \ \oplus = \begin{cases} +, & if \ T_{S \to N} > 0 \ and \ T_{N \to T} > 0 \\ -, & if \ T_{S \to N} > 0 \ and \ T_{N \to T} < 0 \\ -, & if \ T_{S \to N} < 0 \ and \ T_{N \to T} > 0 \\ \infty, & if \ T_{S \to N} < 0 \ and \ T_{N \to T} < 0 \end{cases} \quad and \quad n(I_{S \to T}) = \left\lceil \frac{n(I_S \cap I_N) + n(I_N \cap I_T)}{2} \right\rceil \tag{3}$$

In plain words, in order to compute the inferred trust in a trust path that associates a source user S with a target user T through one intermediate node N, we first compute the weighted sum of the two direct trust associations of S, N and N, T using as weights the number of co-rated items of each direct association, and then apply a sign to the weighted sum according to table 1.

Table 1. Definition of the sign of the inferred trust in a trust path

	$T_{S \to N} \geq 0$	$T_{S \to N} < 0$
$T_{N \to T} \geq 0$	+	-
$T_{N \to T} < 0$	-	∞

The intuition behind this computation is that:

- If user *S* trusts user *N* and user *N* trusts user *T* then it is inferred that user *S* trusts user *T*
- If user *S* does not trust user *N* and user *N* trusts user *T* then it is inferred that user *S* does not trust user *T*
- If user *S* trusts user *N* and user *N* does not trust user *T* then it is inferred that user *S* does not trust user *T*
- If user *S* does not trust user *N* and user *N* does not trust user *T* then inference is not applicable and the length of the supposed path between user *S* and user *T* is considered infinite

The computed value of the inferred trust is a value that lies between the values of the two direct trust associations as indicated in Equation 4 and it is biased towards the value of the direct trust association with the most co-rated items. For example, if $T_{S \to N} = 0,7$ based on 5 co-rated items and $T_{N \to T} = 0,35$ based on 2 co-rated items, then $T_{S \underset{N}{\to} T} = 0,6$. In the same context, if $T_{S \to N} = 0,7$ and $T_{N \to T} = -0,35$, then $T_{S \underset{N}{\to} T} = -0,6$.

$$\min\{T_{S \to N}, T_{N \to T}\} \le T_{S \underset{N}{\to} T} \le \max\{T_{S \to N}, T_{N \to T}\} \tag{4}$$

3.4 Confidence and Uncertainty Properties of Trust Associations

Network evolution is based on individual rating behavior, thus it is reasonable to consider that available structural information defines multiple personalized webs of trust [22]. The *personal web of trust* or *local trust* for a user *S* is given through the set of trust paths originating from *S* and passing through users he or she trusts *directly* or *indirectly*. Figure 2(c) depicts the notion of personal web of trust. Consequently, a user *S* that interacts with other users in the system develops a *subjective belief* of the network. By subjective belief, we mean that probably what a user in the network believes about *S* is different from what another user in the network believes about user *S*. In order to express this subjective notion of trust we set up a confidence model able to respond to the following interrelated questions:

Q1: How confident user *S* feels of his or her opinion about user *T*?
Q2: What is the uncertainty enclosed in user's *S* opinion about user *T*?

Confidence Property
We define as *confidence*, a property assigned to each direct association of the network that expresses the reliability of the association. We make the assumption that confidence is directly related to the number of co-rated items between two users. This assumption indicates that (a) a user's opinion becomes more reliable as additional co-rated items become available and that (b) the reliability of an association between two users may be influenced by the change of the number of co-rated items between other users in the system. For that reason, the more items two users have co-rated, the higher the degree of confidence their association would have. Confidence is applied to

each one of a user's direct associations and it is based exclusively on the user's rating activity. In order to compute the confidence of all direct associations of a user, we initially identify the most confident association in an individual's personal web and then express all confidence values of the remaining direct associations in relation to the identified most confident association. We denote the user with which the most confident association has been created as u_{MAX_CONF}. If I_x is the set of items that user u_x has rated, and $n(I_x)$ is the cardinality of the set I_x, then the confidence $C_{S \to T}$ of the association between the source user S and the target user T is given by equation 5.

$$C_{S \to T} = \frac{n(I_S \cap I_T)}{n(I_S \cap I_{u_{MAX\_CONF}})} \tag{5}$$

Figures 3(a) and 3(b) show how confidence values of direct associations derive from the number of co-rated items between the source user S and the remaining users in the system. The value of the most confident direct association is always equal to 1, while all other direct associations are equal to or less than 1 as depicted in Figure 3(b).

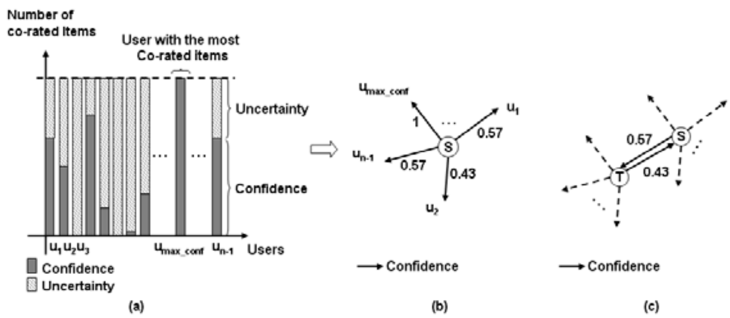

Fig. 3. Confidence Model to Define Uncertainty and Subjectiveness of Trust

Uncertainty Property

The confidence model described earlier can be employed to define *uncertainty* [23]. We define as uncertainty, a property assigned to each direct association of the network that expresses the unreliability of the association. Uncertainty, just like confidence is directly related to the number of co-rated items between two users. This assumption indicates that (a) the uncertainty enclosed to a user's opinion is greater when the number of co-rated items is small and that (b) the uncertainty of an association between two users may be influenced by the change of the number of co-rated items between other users in the system. It becomes obvious that in our model, confidence and uncertainty are contradictory and complementary. Consequently, the more confident one feels about his or her opinion of a user, the less uncertainty is enclosed in his or her opinion of that user and vice versa. Uncertainty $U_{S \to T}$ of the association between the source user S and the target user T is given by equation 6.

$$U_{S \to T} = 1 - C_{S \to T} \tag{6}$$

Confidence and Uncertainty in Trust Paths

Confidence and uncertainty properties may also be assigned to trust paths. We adopt a transitivity rule that facilitates the computation of the confidence between a source user and a target user through a trust path [21, 22]. If we define the set of intermediate nodes in a trust path that associate a source user S with a target user T as $N=\{N_i: i=1, 2, \ldots,k\}$, then the confidence of the trust path is given by Equation 7. Accordingly, the uncertainty assigned to the trust path is given by equation 8.

$$C_{S \underset{N_1 \to \ldots \to N_k}{\to} T} = \left(\left(\left(\left(C_{S \to N_1} \cdot C_{N_1 \to N_2} \right) \cdot \ldots \right) \cdot C_{N_{k-1} \to N_k} \right) \cdot C_{N_k \to T} \right) \quad (7)$$

$$U_{S \underset{N_1 \to \ldots \to N_k}{\to} T} = 1 - C_{S \underset{N_1 \to \ldots \to N_k}{\to} T} \quad (8)$$

Subjectiveness

Since the evolution of personal webs is based on individual rating behavior one would expect that confidence and uncertainty are defined from a user's perspective. Indeed, confidence and uncertainty are *bidirectional* properties. This means that even if two users trust each other as much as what a similarity measure indicates, they do not necessarily have the same confidence in this association. Consider for example, the illustration of Figure 3(c) where there is a direct trust association between user S and user T. Since computation of trust is based on user similarities their associated trust would be the same for both users. However, user S is as much as 0.57 confident about this association, while user T is as much as 0.43 confident about this association. Therefore, our approach is in accordance with the widely accepted position that trust has a subjective notion [23] and reflects the way in which trust is raised in real world social networks.

3.5 Managing Multiple Trust Paths

Since trust inferences are based on traversal paths in a network, it is possible to find *multiple paths* that connect two users. Figure 4 depicts an example in which a source user S is connected to a target user T through two alternative trust paths P_A and P_B.

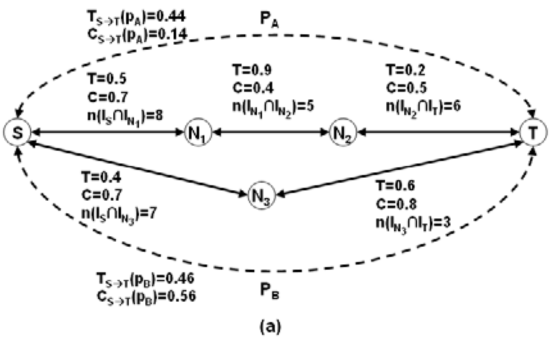

Fig. 4. Illustrating Example of Multiple Trust Paths

Path P_A passes through users N_1, N_2, while path P_B through user N_3. The inferred trust in each of these trust paths is independent of the other. Thus, our trust model needs to define a rule that decides which of these inferred trusts to take into consideration. We describe two approaches for inferring trust when there are multiple trust paths available; the first approach is based on *path composition*, while the other is based on *path selection*. For the following approaches we assume that there are p discrete paths between user S and user T.

Path Composition

The path composition approach tries to combine the values that are inferred by the multiple paths to one single trust value. We distinguish between two methods of composition; *Average Composition* and *Weighted Average Composition*.

- *Average Composition*: We compute the average of all the trust values that are inferred by each of the alternative paths according to Equation 9. Despite the fact that this approach is very cost effective it is considered too naive, because it doesn't take into consideration the confidence of each path.

$$T_{S\to T} = \frac{\sum_{i=1}^{p} T_{\underset{P_i}{S\to T}}}{p} \tag{9}$$

- *Weighted Average Composition*: We compute the weighted average of the trust inferred by the alternative paths, using for weights the propagated confidence of each inferred association between user S and user T, according to Equation 7. This approach is more sophisticated since path confidence is taken into consideration. The final computed trust would be biased to the trust inferred by the most confident path.

$$T_{S\to T} = \sum_{i=1}^{p} \frac{C_{\underset{P_i}{S\to T}}}{\sum_{i=1}^{p} C_{\underset{P_i}{S\to T}}} T_{\underset{P_i}{S\to T}} \tag{10}$$

Path Selection

The path selection approach tries to identify the most confident path among the paths available. We employ two methods of selection, one based on *Maximum Path Confidence* and one based on *Minimum Mean Absolute Deviation (MAD).*

- *Selection Based on Path Maximum Confidence*: Based on the confidence of direct association we can compute the confidence of a path in the network according to Equation 7. Thus, it is possible to compute the confidence of all discrete paths and then to select the one with the highest degree of confidence. Then, we can use only this path to compute the inferred trust between user S and user T.

$$T_{S\to T} = \max\{C_{\underset{P_i}{S\to T}} : i = 1,2,...,p\} \tag{11}$$

- *Selection Based on Minimum Mean Absolute Deviation (MAD)*: It is possible to order the discrete paths that connect user S and user T, according to the Mean Absolute Deviation of their direct associations. We consider absolute deviation to be the difference between the confidence values of two consecutive associations. Once all MAD values are computed for each of the paths available we select the one with the minimum MAD as indicated by Equation 12, where N is the cardinality of nodes in the path p. This path selection method requires that the path comprises of at least 3 users (i.e. $N \geq 3$). The assumption of this approach is that a path would be more confident when consecutive values of confidence introduce smaller instability.

$$T_{S \to T} = \min\{MAD(P_i) : i = 1, 2, ..., p\}, \quad where \quad MAD(P_i) = \frac{\sum_{k=1}^{N-2} \left\| C_{N_k \to N_{k+1}} \right| - \left| C_{N_{k+1} \to N_{k+2}} \right\|}{N-2} \tag{12}$$

4 Experimental Evaluation and Results

In this section we evaluate our method for alleviating the sparsity problem using trust inferences. Our evaluation scenario spans across two dimensions. We first evaluate the *impact of trust inferences* to the sparsity problem and then evaluate the *quality of the recommendations* that are based on the underlying network of direct and inferred associations. The experimental data come from our movie recommendation system named *MRS*. The lowest level of sparsity introduced by the system is 0.972 which is a typical sparsity level for recommendation systems, while ratings range from 1 to 10.

4.1 Trust Inference Impact

Our first objective was to introduce a method that would lead to additional information accessible for recommendation purposes. We have run tests to discover how much more informative or "dense" is the user-item matrix after applying our method of trust inferences. However, since inferences are dependent on user rating activity we first provide an allocation of ratings that correspond to each user. This helps understanding the peculiarities of our network. Figure 5 illustrates the user

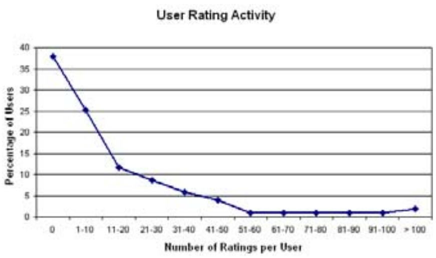

Fig. 5. User Rating Activity

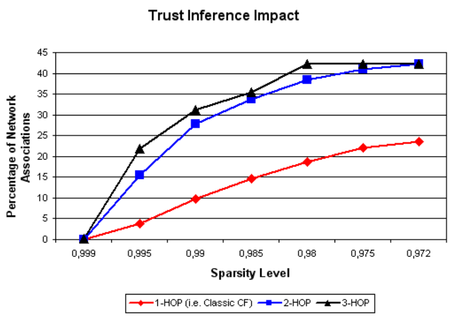

Fig. 6. Impact of Trust Inference for Different Sparsity Levels

rating activity in our recommendation system, which seems to follow a *power law distribution* (Zipf distribution) [24]. There are a few users that have submitted many ratings, some users with normal number of ratings and many users with a few or even no ratings. It is essential to mention that 38% of users have no rating. This means that in our system there are some users for which no information is available, and therefore recommendations are not possible. However, for the rest users, which are members of the underlying social network, our methodology seems to be beneficial.

For our experiments, we define as *k*-HOP CF the method that employs neighbor users that are *k* hops away from the active user. We compute the percentage of user pairs that are feasible in the network when *1*-HOP, *2*-HOP and *3*-HOP CF algorithms are employed and for different sparsity levels. *1*-HOP CF represents the classic CF algorithm, while *2*-HOP CF and *3*-HOP CF represent our trust inference based transitive method for 2 and 3 hops away respectively. According to Figure 6, the percentage of network associations considered by the Classic CF are fewer than these considered by our transitive method. This is consistent with our theory, since Classic CF (*1*-HOP) employs only direct associations, while *2*-HOP CF and *3*-HOP CF apply transitive properties in the network. In addition, it is shown that for sparsity level of 0.972, the *1*-HOP CF considers approximately 24% of the total user pairs, while *2*-HOP and *3*-HOP consider approximately 43% of the total user pairs. It is also demonstrated that after a while *1*-HOP, *2*-HOP and *3*-HOP CF algorithms reach an *upper limit*. This limit is defined by the percentage of users that are inactive in the system, and therefore are not connected to the underlying network. Furthermore, it is depicted that 3-HOP CF has similar results to 2-HOP CF, thus for the recommendation quality experiments we only consider the 2-HOP CF algorithm, which has better time performance.

4.2 Recommendation Quality

If a *prediction* is defined as a value that expresses the predicted likelihood that a user will "like" an item, then a *recommendation* is defined as the list of *n* items with respect to the top-*n* predictions from the set of items available. Thus we can reduce the problem of recommendation quality to the problem of prediction quality for our experiments. More accurate prediction algorithms indicate better recommendations. *Statistical accuracy* and *decision-support accuracy* are the key dimensions on which the quality of a prediction algorithm is usually evaluated.

Statistical Accuracy Metrics

Statistical accuracy metrics evaluate the accuracy of a prediction algorithm by comparing the numerical deviation of the predicted ratings from the respective actual user ratings. Some of them frequently used are *Mean Absolute Error (MAE)*, *Root Mean Squared Error (RMSE)* and *Correlation* between ratings and predictions [19]. As statistical accuracy measure, Mean Absolute Error (MAE) is employed. Formally, if *n* is the number of actual ratings in an item set, then MAE is defined as the average absolute difference between the *n* pairs $< p_h, r_h >$ of predicted ratings p_h and the actual ratings r_h and is given by equation 13.

$$MAE = \frac{\sum_{h=1}^{n} |p_h - r_h|}{n} \tag{13}$$

The lower the *MAE*, the more accurate the predictions are, allowing for better recommendations to be formulated. *MAE* has been computed for Classic CF and for the four variations of our *2*-HOP CF method based on trust inferences. The prediction algorithms are tested for different levels of sparsity over a pre-selected *300-ratings set* extracted randomly by the set of actual ratings. Figure 7 illustrates the sensitivity of the algorithms in relation to the different levels of sparsity applied.

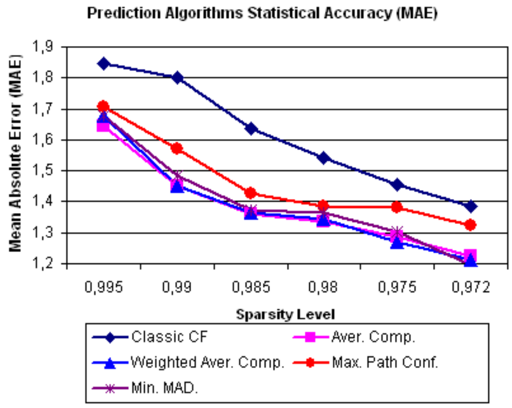

Fig. 7. MAE of the Classic CF and the variations of our CF method of trust inferences for different Sparsity lpevels

As far as statistical accuracy is concerned *2*-HOP CF algorithm outperforms the *1*-HOP Classic CF for all sparsity levels. For typical sparsity levels of recommendation systems, such as 0.975 and 0.98, *2*-HOP CF performs as much as 10.1% and 13.1% better than *1*-HOP CF respectively. In cases that data is extremely sparse, for example when it is equal to 0.99, *2*-HOP CF performs as much as 17% better than *1*-HOP CF. Considering that most of the alternative methods proposed for dealing with the sparsity problem result in recommendation quality degradation, the quality performance of our prediction algorithms is very satisfactory.

Decision-Support Accuracy Metrics

Decision-support accuracy metrics evaluate how effectively predictions help a user to select high-quality items. Some of them frequently used are *reversal rate*, *weighted errors*, *Precision-Recall Curve (PRC) sensitivity* and *Receiver Operating Characteristic (ROC) sensitivity*. They are based on the observation that, for many users, filtering is a binary process. Consequently, prediction algorithms can be treated as a filtering procedure, which distinguishes “good” items from “bad” items.

As decision support accuracy measure, ROC sensitivity is employed. ROC sensitivity is a measure of the diagnostic power of a filtering system. Operationally, it is the area under the receiver operating characteristic (ROC) curve, a curve that plots the sensitivity and the 1-specificity of the test. Sensitivity refers to the probability of a randomly selected "good" item being accepted by the filter. Specificity is the probability of a randomly selected "bad" item being rejected by the filter.

If *PR, AR, QT* denote the predicted rating, the actual rating and a quality threshold respectively, then the following possible cases are defined by the filter for one item

- True Positive (TP) when $PR \geq QT \wedge AR \geq QT$
- False Positive (FP) when $PR \geq QT \wedge AR < QT$
- True Negative (TN) when $PR < QT \wedge AR < QT$
- False Negative (FN) when $PR < QT \wedge AR \geq QT$

For a set of items sensitivity is defined as the True Positive Fraction (TPF) and the 1-specificity as the False Positive Fraction (FPF) where

- $sensitivity = TPF = \frac{tp}{tp + fn}$, where tp, fn is the number of the true positive and the false negative occurrences over the set of items respectively.
- $1 - specificity = FPF = \frac{fp}{fp + tn}$, where tn, fp is the number of the true negative and the false positive occurrences over the set of items respectively.

ROC curve has been computed for different prediction algorithms and for quality thresholds ranging between 1 and 9, while the sparsity level was equal to 0,972. For each prediction we considered a neighborhood of 5 users. The area under the curve represents how much sensitive the prediction algorithm is, so the more area it covers the better for the prediction algorithm. Results are illustrated on Figure 8.

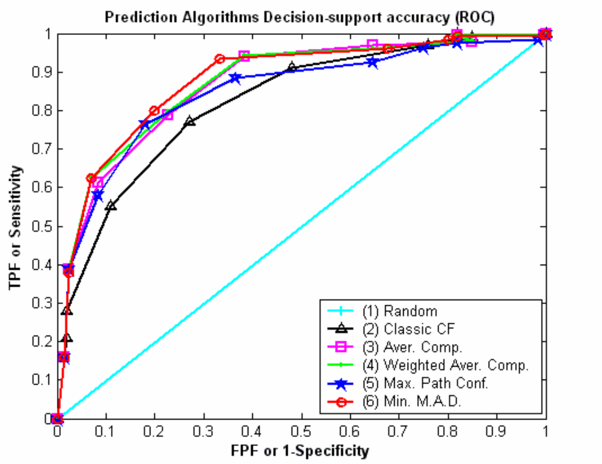

Fig. 8. ROC for the Classic CF and the variations of CF method of trust inferences

As far as decision-support accuracy is concerned the performance of the CF method based on our method of trust inferences is of superior quality than Classic CF prediction algorithms, while there is only slight difference between the accuracy performance of the four variations of our CF method. To obtain a clear view of the overall performance of each algorithm one needs to compute the area under the ROC curve. It is clear from Figure 8 that Classic CF performs much worse than every other algorithm employed based on our method of trust inferences.

5 Conclusions

Sparsity is one of the major aspects that limits the application of the CF method and provokes its success in providing quality recommendation algorithms. In this research, our main objective was to describe a method that is able to provide high-quality recommendations even when information available is insufficient. Our work employs theoretical results of research conducted in areas of social networks and trust management in order to develop a computational trust model for recommendation systems. To deal with the sparsity problem we proposed a method that is based on trust inferences. Trust inferences are transitive associations between users that participate in the underlying social network. Employment of this model provides additional information to CF algorithm and remarkably relaxes the sparsity and the cold-start problems. Furthermore, our model considers the subjective notion of trust and reflects the way in which it is raised in real world social networks. Subjectiveness is defined in terms of confidence and uncertainty properties that are applied to the network associations. We have experimentally evaluated our method according to the impact that trust inferences have to sparsity and according to recommendation quality. Our experimental results indicate that our method succeeds in providing additional information to the CF algorithm while it outperforms the quality performance of the classic CF method. The methodology described is general and may probably be easily adopted to alleviate the sparsity problem in other application areas, especially where underlying social networks can be identified.

References

1. Resnick P. and Varian Hal R. "Recommender Systems (introduction to special section)". Communications of the ACM, 40(3), 1997
2. Herlocker, J., Konstan, J. A., Terveen, L. and Riedl, J. "Evaluating Collaborative Filtering Recommender Systems". ACM Transactions on Information Systems, 22 (1), 2004
3. Herlocker, J. L., Konstan, J. A., and Riedl, J. "Explaining Collaborative Filtering Recommendations". Proc. of the ACM Conference on Computer Supported Cooperative Work, 2002
4. Sarwar, B., Karypis, G., Konstan, J., Riedl J. "Analysis of recommendation algorithms for e-commerce". Proc. of ACM Electronic Commerce, 2000
5. Linden, G., Smith, B., York, J. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering". IEEE Internet Computing, January, 2003

6. Schein, A. I., Popescul, A., Ungar, L. H. “Methods and Metrics for Cold-Start Recommendations". Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002
7. Goldbergh, K., Roeder, T., Gupta, D., and Perkins, C. “Eigentaste: A constant time collaborative filtering algorithm”. Inf. Ret. 4, 2, 133–151, 2001
8. Billsus, D., and Pazzani, M. J. “Learning collaborative information filters”. Proc. of the 15th International Conference on Machine Learning, 46–54, 1998
9. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R. “Indexing by Latent Semantic Analysis”. JASIS 41(6), 1990
10. Hofmann, T., 2003. “Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis”. Proc. of the 26^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003
11. Huang, Z., Chen, H., Zeng, D. “Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering”. ACM Transactions on Information Systems, 22 (1), 2004
12. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. “Item-based Collaborative Filtering Recommendation Algorithms”. Proc. of WWW, 2001
13. Popescul, A., Ungar, L. H., Pennock, D.M., Lawrence, S. “Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments”. Proc. of UAI, 2001
14. Melville, P., Mooney, R. J. and Nagaragan, R. “Content-Boosted Collaborative Filtering for Improved Recommendations”. Proc. of the National Conference of the American Association Artificial Intelligence, 2002
15. Shardanand U. and Maes P. “Social information filtering: Algorithms for automating "word of mouth"”, Proc. of CHI, 1995 -- Human Factors in Computing Systems, 210-217, 1995
16. Allen, R.B. “User Models: Theory, method and Practice”. International Journal of Man-Machine Studies, 1990
17. Ziegler, C.N. and Lausen, G. “Analyzing Correlation Between Trust and User Similarity in Online Communities”. Proc. of the 2^{nd} International Conference on Trust Management, 2004
18. Breese, J. S., Heckerman, D. and Kadie, C. “Empirical analysis of predictive algorithms for collaborative filtering”. Proc. of the 14^{th} Conference on Uncertainty in Artificial Intelligence, 1998
19. Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. “An Algorithmic Framework for Performing Collaborative Filtering”. Proc. of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval, 1999
20. Pearson K. “Mathematical contribution to the theory of evolution: VII, on the correlation of characters not quantitatively measurable”. Phil. Trans. R. Soc. Lond. A, 195, 1-47, 1900.
21. Guha,R., Kumar,R., Raghavan, P., Tomkins, A. “Propagation of Trust and Distrust”. Proc. of WWW2004
22. Ziegler, C.N. and Lausen, G. “Spreading Activation Models for Trust Propagation”. Proc. of IEEE International Conference on e-Technology, e-Commerce, and e-Service, 2004
23. Josang, A. “A Logic for Uncertain Probabilities”. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 9, No. 3, June, 2001
24. Faloutsos, M., Faloutsos P., Faloutsos C. “On Power-Law Relationships of the Internet Topology”. Proc. of SIGCOMM, 1999

25. Papagelis M., Plexousakis D. "Qualitative Analysis of User-based and Item-based Prediction Algorithms for Recommendation Agents". International Journal of Engineering Applications of Artificial Intelligence, vol 18(4), June, 2005
26. Papagelis M., Rousidis I., Plexousakis D., Theoharopoulos E. "Incremental Collaborative Filtering for Highly-Scalable Recommendation algorithms". Proc. of International Symposium on Methodologies of Intelligent Systems, 2005