Propagating Users' Similarity towards improving Recommender Systems

Anna Satsiou and Leandros Tassiulas
Department of Electrical and Computer Engineering
University of Thessaly, Greece
E-mails:(satsioua, leandros)@uth.gr

***Abstract*—In this paper we examine an advanced collaborative filtering method that uses similarity transitivity concepts. By propagating "similarity" between users, in a similar way as with "trust", we can significantly expand the space of potential recommenders and system's coverage, improving also the recommendations' accuracy. While "trust" information might be missing or be misleading and incorrect, "similarity" between two users can be directly calculated using the information from users' item ratings. A recent study observed a strong correlation between trust and preference similarity in online rating systems; therefore it makes sense that transitivity concepts can also be applied to "similarity", much as they are applied to "trust". In contrast to a vast amount of work that seeks to exploit existed social information, like trust, from social networks to improve the recommendation process, we propose the other way round towards the same goal: use similarity transitivity concepts exploiting the rating history of the recommender system's users to lead to the formation of new relationships and even social communities that were not previously existed. We propose a novel similarity propagation scheme to confront the data sparsity problem in recommender systems and evaluate our method over two datasets with different characteristics, exhibiting a much higher recommendation coverage and better accuracy than classical collaborative filtering methods even under very sparse data conditions.**

Keywords*—*recommender systems; similarity transitivity; collaborative filtering; data sparsity; cold-start problem

I. Introduction

Recommender Systems have been extensively used in Internet companies (e.g., Amazon, Ebay, etc) and online social communities (e.g., Epinions, Advogato, etc), as a response to the information overload problem, providing users with reliable information of their particular interests. Due to their commercial value many research communities (cognitive science, data mining, information retrieval, machine learning, etc) have dealt with recommendation techniques.

Collaborative filtering (CF) is the most widely used technique to provide personalised recommendations. It provides recommendations to a given user, exploiting his rating history in order to find users (also known as neighbors) who share similar or highly correlated ratings with him. The basic idea is that the target user will most likely prefer those items that similar users like.

Although CF is such a popular technique, it faces an inherent weakness of recommender systems, namely the data sparsity problem [1]. The available ratings in commercial recommender systems is often less than 1% and thus the ratings matrix is very sparse; therefore in most of the cases there won't be any items rated by both two users, making the similarity comparison between those two impossible. Due to this situation, good recommenders can only be seeked among a small portion of users that can be compared, although many more or even better recommenders may exist in the system. On the other hand, failure of the system to find "good" recommenders will lead to poor quality recommendations.

Recent efforts seeked to alleviate aforementioned weakness by providing trust-based recommendations. A previous study has shown that users tend to prefer recommendations from people they know and trust [2]. In several commercial recommendation systems (e.g., Epinions, Advogato, Filmnet), a user is requested to explicitly state the trust he has on another user, creating a web of trust. By using the trust graph of the recommendation system and the trust transitivity property, it is possible to include many more good recommenders in the recommendation process. Towards this goal, several trust metrics and trust propagation algorithms have been proposed [3]–[6].

Although this approach seems to indeed expand the space of good recommenders for a given user, it is subjected to several restrictions, listed below:

a) Trust information is not provided in all online recommender systems (e.g., cases of MovieLens, Last.fm, Jester, BookCrossing, etc),

b) Users are usually reluctant to express "publicly" the trust they have on other users, on fear of offending or upsetting their less trusted affiliates. For example, consider that user A rates B as trusted with a high trust value , while B rates A with a low trust value. It is very likely that A will be disappointed being aware of that. This fact might have caused the high amount of noise that has been observed in the data of Epinions site [7] (several users pose a high value of trust to others on fear of being misunderstood),

c) Trust information is not always accurate. This might be caused due to above considerations or even system design. For example, Advogato system limits the amount of granted global certifications (for highly-trusted users), and assigns a lot of Observer certificates (low-reputed users); this biases the amount of trusted and untrusted users [5],

d) Trust may not always reveal "similar tastes" but refer to other issues that make a user to trust another one, which may be irrelevant to the recommendation process. For example, A trusts B to share his secrets, but A may like totally different movies than B; therefore A would never ask B's opinion of a movie. It is notable that the tastes of one user's friends may significantly differ.

e) Although trust-based recommendation policies are argued to better deal with the cold-start problem of newcomers than collaborative filtering methods, there are frequently cases when a new user to the system has no established trust relations with other users and thus cannot identify the trusted ones from whom he could ask recommendations, rendering him vulnerable to manipulation.

Considering the above limitations, we propose an alternative way to expand the space of good recommenders and confront the data sparsity problem, without relying on trust information which may be missing or be incorrect: "Similarity" between two users in terms of their preferences can be considered a dimension of trust, which is relevant to the recommendation process. Work in [8] supports this statement, by observing a strong correlation between trust and preference similarity in online rating systems. As such, "Similarity" could also follow the propagation and transitivity properties that apply for trust, and in contrast to trust that usually needs to be explicitly stated by users, similarity can be extracted by users' rating data (e.g., using the Pearson correlation coefficient). By using similarity propagation and transitivity concepts, we can significantly increase the number of personalised recommenders for given items and users and improve the recommendation coverage.

Another promising feature of this strategy is the potential creation of social bonds between users of similar preferences. Many recommendation systems are designed on top of already established social networks (e.g., FilmTrust, Flixter, Epinions, CIAO, etc) and exploit social information, like trust between users, to improve recommendations; under our proposed setup, a social network can be created on top of the recommendation system, i.e., users may discover and make new friends with whom they share similar interests, through the similarity graph of the proposed recommendation system. The theory of homophily [9] arisen by sociology and psychology fundamental research ([10], [11]) indicates that people are attracted and tend to form ties with other people who are similar. More recent research work has also implied that similarity can serve as a strong predictor of friendship [12], [13]. Based on these indications, it would be very meaningful and valuable to connect people that tend to have similar preferences over various items/products/ideas/jokes in the various online systems, and our framework could help towards this direction.

The main contributions of this paper can be summarized as follows:

a) we propose a method to overcome identified limitations of most popular similarity measures, introducing a new similarity measure for the cases the latter fail, e.g., when users have rated limited common items,

b) we introduce "Similarity Factor" which is a function of the similarity between two users and the number of their commonly-rated items. Similarity Factor incorporates in this way the confidence in the predicted similarity; the higher the number of common ratings the stronger the confidence in the similarity prediction and the higher the Similarity Factor,

c) we propose a rating prediction policy that is based on the users' Similarities Factors (see above) as opposed to most popular prediction policies that are based on users' similarities,

d) we propose a novel similarity propagation policy to overcome data sparsity problem and enlarge the space of good recommenders,

e) we evaluate our scheme in two datasets (Movielens and Jester) and show that it can achieve higher accuracy than classical collaborative filtering policies, and the maximum potential rating prediction coverage even when the available rating data are very sparse.

Besides all aforementioned contributions, this paper provides ideas that could be further explored to connect people based on their predicted similarities (direct and transitive) in recommender systems and build social networks on top of them.

The remainder of this paper is organized as follows: Section II summarizes and compares related work and section III describes classic collaborative filtering methods and its limitations. In section IV the proposed framework is described and section V evaluates it over two different datasets. Finally section VI concludes the paper.

II. Related Work

A survey on recommendation systems is given in [14], while [15] provides a survey specifically on collaborative filtering recommendation systems. [16] examines different variations of the most popular collaborative filtering algorithms over MovieLens dataset, presenting certain benefits in terms of prediction accuracy. There have been several other papers seeking to improve the accuracy of the personalised recommendation policies, either by proposing new similarity metrics (e.g. [17]) or filtering methods to eliminate least suitable recommenders for a given item (e.g. [18]). None of these, though, deal with the data sparsity or cold-start problem.

Towards confronting the data sparsity problem, several papers proposed new similarity measures targeting specifically this issue. Work in [19] presents a new similarity metric that uses information from the users' ratings; although the paper reports accuracy improvements compared to most popular similarity measures, it does not exhibit results related to the systems' coverage which is among the most suffering metrics from the cold-start problem. [14] study actually exhibits that the proposed metric in [19] cannot provide full potential coverage even under medium sparsity conditions. Better prediction accuracy but even lower coverage than [19] is achieved by the proposed metric in [20].

Other papers seek to deal with the same problem by exploiting social information, like trust or tags. Related surveys are presented in [21], [22]. Work in [23] exploits social contextual information and integrates it with the user-item matrix based on probabilistic factor analysis to provide recommendations that are shown to outperform collaborative methods, especially when target users have very few ratings. [4] presents FilmTrust, a website that integrates web-based social networking into a movie recommender system and proposes a trust prediction algorithm to appropriately choose the raters (those with the highest trust values) based on whom a recommendation will be given. Several more trust metrics and trust propagation algorithms have been proposed in [24], [25], [6], [3]. A trust prediction and propagation policy considering abstract similarity propagation concepts is described in [26] which concludes that predicting trust is more successful for users that are similar. More than that, work in [27] examines the evolution of trust through time and how it can help in the performance improvement of rating predictions and [28] proposes a method to capture the multi-faceted trust relationships among users of product review sites and exploits them to improve the recommendation process by preferring those raters that share similar trust to the target user on the particular facet in question. Exploitation of existed social information like trust, however, in order to improve a recommender system, is subjected to the limitations reported in the introduction.

Work in [29] proposes to infer trust from a system in order to improve recommendation coverage, instead of explicitly ask users to pose trust values on other ones. Towards this goal, [29] proposes (i) a way to derive the trust between two users that have rated items in common, (ii) a method to compute indirect trust for distant entities based on Subjective Logic and iii) an approach to map trust back into similarity. In our paper we show that we donot need to follow this whole procedure, since transitivity concepts can be directly applied to similarity (implied as a dimension of trust) providing full potential coverage, and gains in prediction accuracy.

Transitive relationships have been also investigated in behavioural networks [30] and in the context of collaborative filtering [31], [32]. [30] attempts to analyze behavioral similarities between pairs of users by using usage traces. Navigational similarities are transformed over distance-like values (although not explained how) and a Floyd-Warshall algorithm is used to compute shortest paths between users, leading to new links throughout transitive relationships. The paper does not investigate the problem of sparsity and cold start users (in that case, users with little evidence on the online system), and fails to achieve better prediction accuracy than standand CF policies for all items but only for items that have been highly rated by users. On the contrary, our scheme achieves higher accuracy compared to CF over all items, and even when the cold start users are the majority in the system.

[31] proposed a threshold-based similarity transitivity method, based solely on commonly chosen items between users, to filter out inaccurate similarities and replace them with transitivity ones; however it does not deal with the presence of cold start users and very sparse data. [32] uses associative retrieval and spreading activation techniques. Although it seeks to alleviate the data sparsity problem, it does not explore the system coverage. Moreover, relationships (direct and transitive) between users are only explored on the basis of the common items they have experienced and not their ratings; thus a lot of information is lost. It is very common that users who experience the same items, rate them very differently and thus their similarity varies.

In this paper we propose a novel similarity propagation algorithm towards confronting the data sparsity problem. Our scheme overcomes the limitations of aforementioned transitive and other schemes, as explained above, and succeed in achieving higher accuracy and much higher coverage of classical collaborative policies.

III. Recommendation Policies and their Limitations

In this section we describe some baseline and most popular collaborative filtering recommendation policies and their limitations.

A. Non-personalised algorithm

A baseline recommendation policy is the non-personalised one, according to which a predicted rating for an item i for a user A is given by computing the item's average rating over all users in the system, as shown in:

$$p_{A,i} = \frac{\sum\limits_{u \in T_i} r_{u,i}}{|T_i|}, \tag{1}$$

where T_i is the total number of users that have rated item i, and $r_{u,i}$ is the rating of user u to item i.

This policy does not account for the individual preferences of user A and thus fails to satisfy personalised needs.

In the following we describe the most popular personalised recommendation policy that was first introduced by Grouplens and is based on a collaborative filtering method.

B. Collaborative-filtering personalised algorithm (CF)

Under this policy a subset of suitable users, called neighbors, are chosen based on their similarity to the user in question (user A in this example) and a weighted aggregate of their ratings is calculated to provide predictions for this user.

The most appropriate similarity measure between two users A and B has been shown by [16] to be the Pearson correlation coefficient (PC), which can be calculated based on the users' ratings of at least two common items, given by:

$$S_{A,B} = \frac{\sum\limits_{i \in R_{A,B}} (r_{A,i} - \overline{r_A})(r_{B,i} - \overline{r_B})}{\sqrt{\sum\limits_{i \in R_{A,B}} (r_{A,i} - \overline{r_A})^2 \sum\limits_{i \in R_{A,B}} (r_{B,i} - \overline{r_B})^2}}, \tag{2}$$

where $r_{A,i}$ is the rating of user A to item i, $\overline{r_A}$ is the average of the ratings given by user A and $R_{A,B}$ is the set of all commonly rated items by users A and B. As it can be seen, $S_{A,B}$ can only be calculated on the basis of common ratings among users A and B, while other ratings are not considered. $S_{A,B}$ is ranging from [-1,1]; similar users are considered only those with similarity above zero.

In order to determine the neigbors of user A that will be involved in the rating prediction of item i, two techniques are usually used, i.e., correlation-thresholding that chooses as neighbors the users with a similarity value with user A above a certain threshold and best-n-neighbors that chooses the n users with the highest similarity with user A.

When the set of neigbors of user A, N_A, is determined, the predicted rating for an item i for user A is given either by the average of the neighbors' ratings, either by a weighted sum of their ratings and most commonly by an adjusted weighting aggregation, given by the following formula (eq.3) that computes average deviations of neigbors's ratings from their mean rating. Latter policy is based on the fact that users may have different rating distributions centered around different points.

$$p_{A,i} = \overline{r_A} + \frac{\sum\limits_{u \in N_A} (r_{u,i} - \overline{r_u}) S_{A,u}}{\sum\limits_{u \in N_A} S_{A,u}}, \tag{3}$$

There are several variations of Pearson correlation in eq.2, like the constrained Pearson correlation that uses midpoints instead of mean rates and Spearman rank correlation that uses ranks instead of ratings; a study in [16] has shown that they have similar performances. Another comparison study reports on the relation between PC and Spearman correlation [33].

C. Limitations of most popular similarity measures and prediction algorithms

Although PC and its variations have been proved to be successful in many studies as similarity measures, they face significant limitations. As already mentioned, PC (and its variations) can only be calculated for users that have rated at least two common items. In cases of new users in the system that have rated very few items, PC fails to indicate sufficient neighbors, and sometimes none at all. This is getting more critical in systems with sparse data.

Moreover, PC cannot be calculated in cases of users that have provided the same ratings in all items they have rated so far (flat ratings); such a case is for example a user that has rated three items in total, and each one with 4, i.e, his rating vector is (4,4,4). In these cases, the denominator of eq.2 goes to zero.

Similar limitations face the next effective and widely-used correlation measure known as vector or cosine similarity given below:

$$S_{A,B} = \frac{\sum\limits_{i \in R_{A,B}} r_{A,i} * r_{B,i}}{\sqrt{\sum\limits_{i \in R_{A,B}} r_{A,i}^2 * \sum\limits_{i \in R_{A,B}} r_{B,i}^2}}, \tag{4}$$

Although cosine similarity can be calculated in cases of users with just a single common item, in that cases it takes the value of 1 irrespective of the differences in individuals' ratings. The same holds in cases of rating vectors in the same line, i.e., (1,1,1), (5,5,5).

On the other hand, neither the aforementioned similarity measures nor the prediction algorithms consider the amount of confidence on the similarity calculation. When two users have rated only very few items in common, their similarity calculation using these ratings cannot be as accurate as if they have rated a high number of common items.

IV. PROPOSED RECOMMENDATION POLICY

In the following we describe the various modules of our proposed recommendation policy, CF-ADV.

A. An alterantive similarity measure to overcome limitations of the most popular ones

Considering above limitations of most popular similarity measures, we propose to use an alternative similarity measure for the cases that PC fails, e.g., when the two users have rated very few items in common, i.e., below a threshold T, or one of them or both have flat ratings. For all other cases the classic PC will be used. The proposed alternative similarity measure (AS) is given by:

$$S_{A,B} = \frac{\sum\limits_{i \in R_{A,B}} f(r_{A,i}, r_{B,i})}{|R_{A,B}|}, \tag{5}$$

where $f()$ is given by:

$$f(r_{A,i}, r_{B,i}) = \begin{cases} \frac{1}{d_i+1}, & \text{if } (r_{A,i} > r_m \text{ and } r_{B,i} > r_m) \\ \frac{1}{d_i+1}, & \text{if } (r_{A,i} \le r_m \text{ and } r_{B,i} \le r_m) \\ \frac{0.5}{d_i+1}, & otherwise \end{cases} \tag{6}$$

d_i is the absolute distance between the ratings of users A and B for the item i and r_m is the median value of the system's rating scale, e.g., for a [1,5] rating scale, median is given by $(1+5)/2 = 3$.

When the users' ratings of the same item are in accordance, i.e., they are both above median value or both equal or less than median value, similarity is increased by a factor of $1/(d_i+1)$, otherwise similarity is increased by a factor of $0.5/(d_i+1)$. Similarity is increased even in the latter case (although less than the former case), since the fact that both users have rated the same item is by itself an indicator of similarity (it is reasonable to infer that users that rate (and thus have experienced) common items, share some common tastes).

Proposed similarity can take any value between [0,1].

B. Similarity Factor

Another shortcoming of classic personalised recommendation policy is the fact that it does not consider the confidence level in the similarity calculation. When the co-rated items between two users are very few, similarity calculation cannot be very accurate. In order to give more weight to neigbors that are similar based on a bigger set of co-rated items, we define the similarity factor that is given by:

$$SF_{A,B} = \begin{cases} S_{A,B} * |R_{A,B}| & \text{if PC is used} \\ S_{A,B} * |R_{A,B} - 0.1| & \text{if AS is used} \end{cases} \tag{7}$$

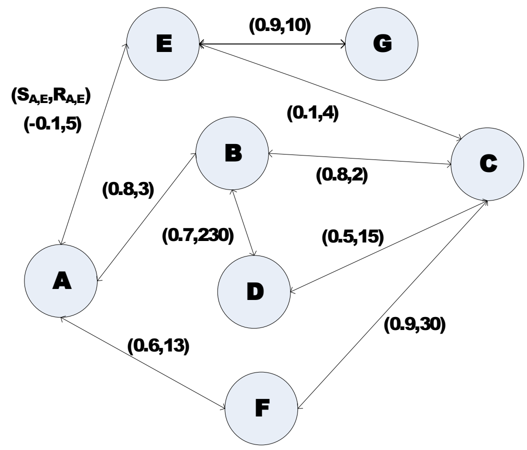

Fig. 1: Similarity Graph.

The more the items that are co-rated between the users, the higher the validity of the similarity calculation and the higher the similarity factor between them. As explained above, AS is used only in cases that PC fails.

C. A rating prediction policy based on Similarity Factor

According to the proposed prediction policy, the predicted rating for an item i for a user A is given by a variation of policy in eq.3, where the weights are the similarity factors and not the similarities of the users, as shown below:

$$p_{A,i} = \overline{r_A} + \frac{\sum\limits_{u \in N_A} (r_{u,i} - \overline{r_u}) SF_{A,u}}{\sum\limits_{u \in N_A} SF_{A,u}}, \tag{8}$$

In this way, we give more weight to the rating of those users who have not only high similarity but also high amount of co-rated items with the user in question.

D. Similarity Graph, Friendship Graph and Transitivity concepts

If we consider that each two users in a recommender system that have rated common items are linked, then a similarity graph can be created like in Fig.1 where each link has a double weight $(S_{A,B}, R_{A,B})$ where $R_{A,B}$ gives the total number of common ratings based on which $S_{A,B}$ has been calculated and signifies the accuracy of the similarity prediction. The higher $R_{A,B}$ is, the more accurate $S_{A,B}$ is.

Let's suppose that user C has rated an item of interest for user A. In Fig.1 we can see that user A does not have common ratings with user C and thus their similarity cannot be predicted. Therefore we cannot know whether the rating of user C is useful to user A. By using similarity transitivity though, we could predict $S_{A,C}$ through intermediary similar to A users, i.e., users B and F. In this way we can significantly increase the number of potential recommenders to a given user and create new links in the similarity graph. If we cut the links with negative similarity, we can say that we create a predicted

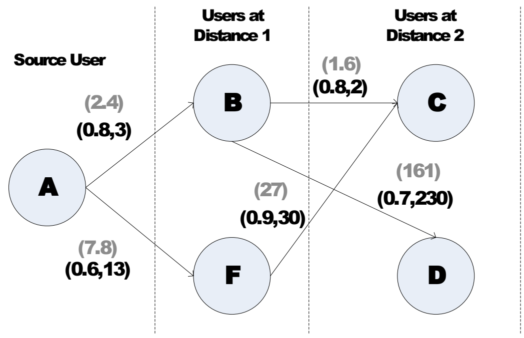

Fig. 2: Friendship Graph of User A for $H = 2$.

friendship graph , based on previous work that has shown that similarity can serve as a strong predictor of friendship [12], [13]. Such graphs could be used by recommender systems to create social networks on top of them and connect unknown, though similar people.

If we then want to find recommenders for a source user A over an item i, we first search over his direct friends (similar users). If none of his direct friends has rated the item i, we use the following proposed similarity propagation policy of two steps, inspired by transitivity schemes that have been proposed for "trust" [4], [25], in order to find good recommenders for user A.

A. Transform predicted friendship graph (i.e., similarity graph without links of negative similarity) into a directed acyclic graph, destroying any cycle in order to visit each node just once starting from the source node (linear time complexity) and then order users based on shortest path distance from the source user, setting the similarity propagation horizon to H and keeping nodes only till that distance. For example, for $H = 2$, the graph of Fig.1 would look like Fig.2, where the double weight of each link can be replaced by a single weight given by the Similarity Factor (in grey), which as explained previously is a linear function of the similarity and the common ratings between two users.

B. Calculate the Similarity Factors of all users until distance H, based on the proposed propagation formula described below, according to which the similarity factor of a user X with a given source user A in a distance $D \leq H$ is given by:

$$SF_{A,X} = \frac{\sum\limits_{u \in P} (SF_{A,u} * SF_{u,X})}{\sum\limits_{u \in P} SF_{A,u}}, \tag{9}$$

where P is the set of all users at distance $D - 1$ that are directly linked with user X (predecessors). Under this policy, the similarity of a user at distance D depends only on the similarities of his predecessors.

For example, in Fig.2 the similarity factor of A with C can be calculated as follows: $SF_{A,C} = \frac{SF_{A,B} * SF_{B,C} + SF_{A,F} * SF_{F,C}}{SF_{A,B} + SF_{A,F}} = 21.02$. In case of $H = 3$, the system first calculates the similarity factors of users a distance 2 and then at distance 3, and the same procedure holds for bigger values of H.

V. Performance Evaluation

A. DataSets

In order to validate our proposed method, we apply it in two datasets with different characteristics described below:

MovieLens

MovieLens is a web site that helps people find movies to watch. The Movielens dataset used in this paper has been collected by GroupLens Research and is available in http://movielens.org. This data set consists of 100.000 discrete ratings (1 to 5) from 943 users on 1682 movies. Each user has rated at least 20 movies.

JESTER

Jester is a Joke Recommender System. In this paper we use a subset of the Jester available set in http://www.ieor.berkeley.edu/ goldberg/jester-data/. This subset consists of 24.761 continuous ratings (-10,00 to +10,00) from 1001 users on 100 jokes. Each user has rated at least 16 items.

We can create more sparse datasets from above, by deducing ratings from certain users, simulating in this way cold start users, as we will explain in section V-C.

B. Performance Metrics

The effectiveness of a rating prediction policy is judged by both its accuracy and coverage. In terms of accuracy there are statistical accuracy metrics that compare the user actual ratings with the rating predictions by the recommendation policy, and decision-support/classification accuracy metrics that assess how efficiently a policy recommends items to users.

The most commonly used statistical accuracy metric is Mean Absolute Error (MAE) exhibiting the absolute deviation between actual and predicted ratings by the recommnedations system, given by:

$$MAE = \frac{\sum_{i=1}^{t} |r_{A,i} - p_{A,i}|}{t}, \qquad (10)$$

where t is the total number of predictions.

Since the numerical rating scales of MovieLens and Jester are different, we use Normalised Mean Absolute Error (NMAU) in order to compare results among the different datasets. NMAU is defined as follows:

$$NMAE = \frac{MAE}{(r_{max} - r_{min})}, \qquad (11)$$

where r_{max} and r_{min} are the maximum and minimum rating values that can be given in the system, respectively. In a [1,5] rating scale system, $r_{max} = 5$ and $r_{min} = 1$. MAE and NMAE equally weights every error in the rating prediction, ignoring the fact that some users have rated many more items than some others. Absolute User Error (MAUE) and Normalized Mean Absolute User Error (NMAUE) give equal weight to each user by first computing the MAE (NMAE respectively) for every single user and then averaging over all users.

TABLE I: Classification Matrix.

Actual/Predicted	$PredictedValue <= 3$	$PredictedValue > 3$
$Rating <= 3$	TN	FP
$Rating > 3$	FN	TP

We further use a classification accuracy metric that determines the propability with which a randomly-selected item is correctly classified under those of interest to certain users (recommended) or those that are not (not recommended). Here we set the classification filter at a prediction of 3 (i.e., consider that all items with predicted ratings above 3 will be recommended, while all items with predicted ratings below or equal to 3 will not be recommended). In Table I we present the classification matrix. True Positive (TP) are the cases that were correctly classified as "high rated" and True Negative (TN) are those that were correctly classified as "low rated". False Negative (FN) are the high-rated items that were wrongly rejected by the filter (not recommended), while False Positive (FP) refers to low rated items that were though falsely recommended to users. Therefore, the accuracy of the recommendation system with a classification filter set at 3 is given by:

$$AC_3 = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (12)$$

Last, we measure the coverage of the recommendation policy which refers to the percentage of all users' ratings for which the given policy can provide a prediction (COV). COV should remain as high as possible. In sparse datatasets where a large portion of cold start users exist, coverage is an important metric for the efficiency of the recommendation policy, since a lot of ratings can be hardly predictable. It is common that a recommendation policy can better predict ratings from users with a long history of ratings (old users) than for cold-start users. In order to account for different types of users in the system, we calculate average user coverage (UCOV), by averaging the coverage of each user in the system. In a dataset of 101 users, where the one user is old with 200 ratings and the rest 100 are cold start users with two ratings each, if the system predicts all the ratings of the old user but fails to predict a single rate from the cold-start users, COV would be 50%, while UCOV would be 0.9%, a much better indicator of the real situation, since under this case the recommendation policy cannot provide recommendations to any user but one (the old one).

C. Results

In this section we compare our proposed recommendation policy CF-ADV with the most popular collaborative filtering method as described in section III, CF, and the baseline non-personalised algorithm NP. "H" in CF-ADV-H denotes the depth of the predicted friendship graph that is used for the recommendations. H2 exploits the friends of friends, H3 exploits even the friends of friends of friends, etc. For both

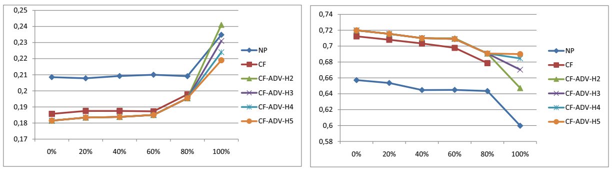

Fig. 3: (a) $NMAE$ and (b) AC_3 for MovieLens Datafile

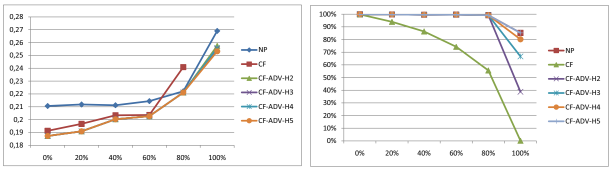

Fig. 4: (a) $NMAUE$ and (b) $UCOV$ for MovieLens Datafile.

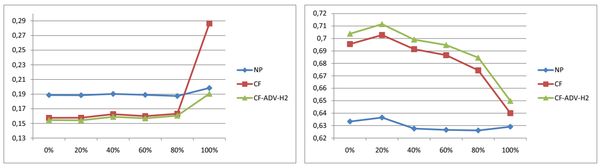

Fig. 5: (a) $NMAE$ and (b) AC_3 for Jester Datafile

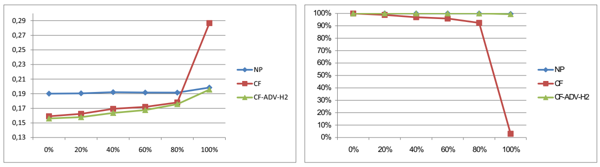

Fig. 6: (a) $NMAUE$ and (b) $UCOV$ for Jester Datafile.

CF and CF-ADV only neighbors with similarity above zero are considered in the recommendation process. For CF-ADV, parameter T was set to 3 (please refer to section IV).

The comparison is made on the two datasets described in section V-A. As can be seen, both datasets consist of users who have provided a satisfying amount of ratings. In reality, however, there is always a mix of new to the system users, known as cold start users, and older users (i.e., users who have provided a lot of ratings). In order to provide our results on realistic conditions, we simulate different percentage of cold start users in the system with a number of ratings uniformly distributed between 1 and 4. We simulate such users by randomly deleting ratings of randomly chosen users from the dataset in order to fulfill aforementioned conditions.

The cold-start problem is an example of the data sparsity situation under real conditions and one of the great difficulties faced by a recommendation system; under such cases it is very difficult to find similar users due to lack of sufficient information and the coverage of the personalised recommendations drops significantly. In this section we show that CF-ADV can achieve full potential coverage even when cold start users are the majority in the system.

In order to evaluate the policies we use 80% ratings of each user for training and 20% for testing and calculate $NMAE$, $NMAUE$, AC_3 and $UCOV$ for each dataset and for different percentage of cold start users ranging from 0% to 100% with a step of 20%.

In Table II we can see the number of identified friendships (pairs of users with similarity above zero) for CF and CF-ADV-H2 policies and different percentage of cold start users in the system for the Movielens and Jester training sets. As we can see, with our policy we can significantly increase the number of friendships in the system, even for a small similarity progagation horizon in the friendship graph ($H = 2$), and thus the space of potential recommenders for a given user and item. The sparser the dataset, the higher the increase in friendships with CF-ADV-H2. For Movielens and 100 % cold start users, friendships are approximately 2000 times more than CF policy and for Jester they are 700 times more. As we can see from Figs. 3-6, this increase in friendships leads to much higher recommendation coverage than CF and accuracy improvements.

In Figs. 3-6 we see that the more the cold start users in the system the lower the performance of all recommendation policies; however, CF-ADV-H2 achieves better prediction accuracy than CF for all the percentage of cold start users in the system and more importantly it provides full coverage even when cold start users are 80% of all system users in MovieLens and even 100% in Jester. CF and CF-ADV policies exhibit a better performance in Jester than in Movielens, because Jester is a more dense dataset. However, the more sparse the available data, the higher the CF-ADV coverage gains over CF.

Moreover, even for Movielens and 100% cold start users if we use a higher depth in the predicted friendship graph we can achieve a higher coverage and a better prediction accuracy as well. With CF-ADV-H5, full potential coverage is achieved. Full potential coverage refers to the maximum coverage that can be achieved given the training and testing datasets; if for example, there are items in the testing set for which there is not a single rating in the training set, no prediction can be given for these items neither by NP or CF policies.

For the Movielens dataset, accuracy results are not exhibited for CF (Figs. 3-4) for the case of 100% cold start users, since in this case the recommendation coverage is almost zero (0,1754%) due to lack of enough identified friendships.

NP can provide full potential coverage for all cases of cold start users, however, its prediction accuracy is much lower than CF and CF-ADV policies, since it cannot provide personalised recommendations. CF-ADV can provide full coverage as NP

TABLE II: Number of friendships (pairs of similar users)

	Movielens		*Jester*	
	CF	CF-ADV-H2	CF	CF-ADV-H2
0%	246245	444153	348161	500500
20%	158843	444094	247787	500498
40%	91273	443643	162470	500487
60%	42847	441987	98355	500406
80%	12560	435779	46863	500104
100%	30	60825	635	453098

policy with a much higher accuracy.

VI. Conclusions and Future Work

In this paper we explored a way to extend the space of potential recommenders for a given user, proposing a novel similarity propagation algorithm that is based on similarity transitivity concepts. As a result, our proposed policy can achieve better prediction accuracy and much higher coverage than most popular CF policy, even when cold start users are the majority in the system.

Our similarity transitivity policy could be applied to any proposed similarity measure. In this paper we chose the most popular one, Pearson correlation, combined with a new metric for the cases that Pearson correlation fails.

Besides providing improvements in the recommendation process, the concept of similarity propagation could be further explored and used to build social networks on top of online systems between unknown users, who though exhibit similarities in choosing or rating items. This is also supported by several sociological and phycological studies which indicate that people tend to form ties with other people who are similar.

Acknowledgment

The authors acknowledge the support of this work by the research program CROWN, through the Operational Program Education and Lifelong Learning 2007-2013 of NSRF, which has been co-financed by EU and Greek national funds.

References

[1] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," *Proc. of the 32nd international ACM SIGIR conf. on Research and development in information retrieval*, pp. 203–210, 2009.

[2] K. Swearingen and R. Sinha, "Beyond algorithms: An hci perspective on recommender systems," *ACM SIGIR Workshop on Recommender Systems*, vol. 13, no. 5-6, pp. 393–408, 2001.

[3] P. Massa and P. Avesani, "Trust metrics in recommender systems," *Computing with Social Trust*, pp. 259–285, 2009.

[4] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," *Proceedings of the 4th international conference on Trust Management*, pp. 93–104, 2006.

[5] P. Massa and K. Souren, "Trustlet, open research on trust metrics," *Scalable Computing: Practice and Experience, Scientific International Journal for Parallel and Distributed Computing*, pp. 341–351, 2008.

[6] J. O'Donovan and B. Smyth, "Trust in recommender systems," *Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 167–174, 2005.

[7] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," *Proc. of the fourth ACM international conference on Web search and data mining*, pp. 287–296, 2011.

[8] J. G. C. Ziegler, "Investigating interactions of trust and interest similarity," *Decision Support Systems*, vol. 43, no. 2, pp. 460 – 475, 2007.

[9] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[10] E. Berscheid, "Interpersonal attraction," *The Handbook of Social Psychology*, pp. 413 – 484, 1984.

[11] T. M. Newcombe, "The acquaintance process," *NY: Holt, Rinehart, and Winston.*, 1961.

[12] C. Jensen, J. Davis, and S. Farnham, "Finding others online: reputation systems for social online spaces," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 447–454, 2002.

[13] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 6, 2000.

[14] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutierrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, no. 0, pp. 109 – 132, 2013.

[15] S. R. M. Hameed, O. Jadaan, "Collaborative filtering based recommendation system: A survey," *International Journal on Computer Science and Engineering*, pp. 859–876, 2012.

[16] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230–237, 1999.

[17] J. Bobadilla, F. Serradilla, and J. Bernal, "A new collaborative filtering metric that improves the behavior of recommender systems," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 520 – 528, 2010.

[18] F. Ortega, J.-L. Sanchez, J. Bobadilla, and A. Gutierrez, "Improving collaborative filtering-based recommender systems results using pareto dominance," *Information Sciences*, vol. 239, no. 0, pp. 50 – 61, 2013.

[19] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Inf. Sci.*, vol. 178, no. 1, pp. 37–51, Jan. 2008.

[20] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225 – 238, 2012.

[21] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," *Computer Communications*, vol. 41, no. 0, pp. 1 – 10, 2014.

[22] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artif. Intell. Rev.*, vol. 37, no. 2, pp. 119–132, Feb. 2012.

[23] H. Ma, T. C. Zhou, M. R. Lyu, and I. King, "Improving recommender systems by incorporating social contextual information," *ACM Trans. Inf. Syst.*, vol. 29, no. 2, pp. 9:1–9:23, Apr. 2011.

[24] J. A. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, College Park, MD, USA, 2005.

[25] P. Massa and P. Avesani, "Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers," *International Journal on Semantic Web and Information Systems*, pp. 39–64, 2007.

[26] P. Borzymek, M. Sydow, and A. Wierzbicki, "Enriching trust prediction model in social network with user rating similarity," *Proceedings of the International Conference on Computational Aspects of Social Networks*, pp. 40–49, 2009.

[27] J. Tang, H. Gao, H. Liu, and A. D. Sarma, "etrust: understanding trust evolution in an online world," *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 253–261, 2012.

[28] J. Tang, H. Gao, and H. Liu, "mtrust: discerning multi-faceted trust in a connected world," *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 93–102, 2012.

[29] G. Pitsilis and S. J. Knapskog, "Social trust as a solution to address sparsity-inherent problems of recommender systems," *Workshop on Recommender Systems and The Social Web*, pp. 33–40, 2012.

[30] I. Esslimani, A. Brun, and A. Boyer, "From social networks to behavioral networks in recommender systems," *Proceedings of the IEEE International Conference on Advances in Social Networks Analysis and Mining*, pp. 143 – 148, 2009.

[31] F. Xie, Z. Chen, H. Xu, X. Feng, and Q. Hou, "Tst: Threshold based similarity transitivity method in collaborative filtering with cloud computing," *Tsinghua Science and Technology*, vol. 18, no. 3, pp. 318 – 327, June 2013.

[32] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116 – 142, Jan. 2004.

[33] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87–93, 2011.