A Unified Framework for Link and Rating Prediction in Multi-modal Social Networks

Panagiotis Symeonidis

Department of Informatics,
Aristotle University,
Thessaloniki, 54124, Greece
Email: symeon@csd.auth.gr

Eleftherios Tiakas

Department of Informatics,
Aristotle University,
Thessaloniki, 54124, Greece
Email: tiakas@csd.auth.gr

Yannis Manolopoulos

Department of Informatics,
Aristotle University,
Thessaloniki, 54124, Greece
Email: manolopo@csd.auth.gr

Abstract:

Multi-modal Social Networks (MSNs) allow users to form *explicit* (by adding new friends in their network) or *implicit* (by similarly co-rating items) social networks. Previous research work was limited either to the prediction of new relationships among users (i.e. Link Prediction problem) or to the prediction of item ratings (i.e. Rating Prediction problem and Item Recommendations). Recent link prediction methods infer future relationships among users, by also exploiting information from their implicit networks (i.e. user-item rating network). On the other hand, Rating Prediction methods predict the user's rating behavior on items, by also exploiting information from their explicit network (i.e. friendship network).

In this paper, we develop a framework to incorporate both research directions into a unified model. We extend our Social-Union algorithm, which initially focused on the rating prediction problem, in order to be applied also on the link prediction problem. Social-Union combines similarity matrices derived from heterogeneous (unipartite and bipartite) explicit or implicit MSNs. Moreover, we propose an effective weighting strategy of MSNs influence based on their structured density. We also generalize our model for combining multiple social networks. We perform an extensive experimental comparison of the proposed method against existing link and rating prediction algorithms, using synthetic and two real data sets (Epinions and Flixter). Our experimental results show that our Social-Union framework is more effective in both rating and link prediction.

Keywords: data mining;social network mining;link prediction;Item Recommendation.

1 Introduction

Multi-modal Social networking sites, like Epinions and Flixter, have attracted huge attention after the widespread adoption of Web 2.0 technology. In such systems, people often belong to multiple explicit or implicit social networks because of different interpersonal interactions. For example, in Epinions and Flixter, people add each other as friends constructing a large unipartite friendship network. However, besides the explicit friendship relations between the users, there are also other implicit relations. For example, users can co-comment on products and they can co-rate products.

There are two main research directions on MSNs. The first concerns the link prediction task, whereas the second refers to the rating prediction and item recommendation task. In the following, we present the motivation of our work in both directions, i.e. the link and rating prediction.

There is a little work in link prediction from multiple social networks. Recently, Lu et al. (2010) proposed supervised link prediction using multiple heterogeneous sources (i.e. auxiliary networks) of information. However, they have not exploited the ability of their model to take into account features other than path counts.

There are many studies (Goldbeck, 2005; Jamali and Ester, 2010; Vasuki et al., 2010) on rating prediction and item recommendation from two or more social networks. Recently, Vasuki et al. (2010) proposed affiliation/group recommendations based on the friendship network among users, and the affiliation/group network between users and groups. However, as they mentioned their method focuses only on path counts and does not exploit other features and network characteristics which can be informative for link formation (i.e.local graph characteristics). Moreover, Yuan et al. (2009) proposed a linear method to integrate explicit social relationships into Collaborative Filtering methods. However, they did not use a weighting strategy of MSNs influence based on their structured data density.

1.1 Contribution

In this paper, we develop a framework to incorporate both research directions into a unified model. We extend our Social-Union algorithm (Symeonidis et al., 2011), which initially focused on the rating prediction problem, in order to be applied also on the link prediction problem. In particular, we decompose the main parts of Social-Union in order to support the link prediction task. Social-Union framework takes into account the local and global characteristics of the graphs such as graph density, user's profile density, nodes structure etc. Moreover, we present a well-defined framework for combining heterogeneous social networks, i.e. unipartite and bipartite networks. It is obvious that not all social networks contribute equally

or contain valuable information. In addition, even though a social network is informative, particular features may be irrelevant and noisy for a specific user. For these reasons, we propose an effective automatic weighting strategy of the social networks influence based on their structured density. In particular, we take into account the local (i.e. user's profile density) and global (i.e. network's density) characteristics of multi-modal social graphs. Based on these characteristics, for each target user we analogously calibrate the influence of each social network. For example, a user could have very few friends in the friendship network, but many interactions in co-commenting or co-rating products (i.e. user-items rating network). In such a case, the weighting strategy of our model promotes the information given by the user-item rating network. Finally, we generalize our model for combining multiple social networks. In particular, our model can incorporate many unipartite (e.g. user-user) or bipartite (e.g. user-item) social networks.

This new Social-Union framework extends our previous work (Symeonidis et al., 2011) as follows:

- We extend our framework in order to provide either friend recommendations (i.e. link prediction) or item recommendations (i.e. rating prediction) based on the user's profile characteristics. That is, if a user has few friend and high interaction with other users on co-rating products, we can exploit the information from the user-item rating network to leverage the friend recommendation task. In contrast, if the user has many friends and low interaction with other users on co-rating products, we can exploit the friendship network to leverage the item recommendation task.
- We propose a new algorithm (i.e. Social-Union-Link-Prediction) which is suitable for link prediction and we present a new toy example that shows how our Social-Union framework can leverage both the rating and the link prediction tasks. Moreover, we provide details about the complexity of the proposed algorithms and we discuss in details about other implementation issues.
- We have experimentally compared our method with both state-of-the-art rating and link prediction algorithms, following two different experimental protocols which are suitable for each different recommendation task (i.e. friend and item recommendation). Moreover, we have also experimentally tested the performance of other aggregation functions (i.e. AggrMin, AggrMax and AggrAvg) for the combination of similarities matrices.

The rest of this paper is organized as follows. Section 2 summarizes the related work, whereas Sections 3 and 4 present how link and rating prediction is performed from bipartite and unipartite graphs, respectively. Our proposed algorithm, which is based on heterogeneous social networks,several extensions, complexity and other implementation issues are described in Section 5. Experimental results are given in Section 7. Finally, Section 8 concludes this paper.

2 Related work

In this Section, we review related work on link and rating prediction.

2.1 Link Prediction

The research area of link prediction in unipartite social networks, tries to infer which new interactions among members of a social network are likely to occur in the near future. There are two main approaches (Liben-Nowell and Kleinberg, 2003) that handle the *link prediction* problem. The first approach is based on local features of a network, focusing mainly on the nodes structure. There is a variety of local similarity measures (Liben-Nowell and Kleinberg, 2003) such as Adamic and Adar index (Adamic and Adar, 2005), Friend of a Friend (FOAF) algorithm (Chen et al., 2009), Preferential attachment (Liben-Nowell and Kleinberg, 2003) etc. The second approach is based on global features, detecting the overall path structure in a network. There is a variety of global approaches (Liben-Nowell and Kleinberg, 2003) such as Random Walk with Restart algorithm (Pan et al., 2004), SimRank algorithm (Jeh and Widom, 2002), Katz status index (Katz, 1953) etc.

Besides the aforementioned link prediction algorithms that are based solely on single-type graph structure, there are also other methods which exploit also other data sources such as messages among users, co-authored paper, common tagging etc. For instance, Gue et al. (2009), proposed a novel user interface widget for providing users with recommendations of people. Their people recommendations were based on aggregated information collected from various sources across IBM organization (i.e. common tagging, common link structure, common co-authored papers etc.). Chen et al. (2009) evaluated four recommender algorithms (Content Matching, Content-plus-Link, FOAF algorithm and, SONAR) to help users discover new friends on IBM's OSN. Recently, Lu et al. (2010) proposed supervised link prediction using multiple heterogeneous sources (i.e. auxiliary networks) of information. However, as they mentioned their method focuses only on path counts and does not exploit other features and network characteristics which can be informative for link formation (i.e. local graph characteristics).

2.2 Rating Prediction

In this Section, we review related work on item recommendation in social networks. We first review memory-based approaches in collaborative filtering (CF), which have been used for recommendation in bipartite social networks (i.e. user-item networks). The GroupLens system (Resnick et al., 1994) implemented a CF algorithm based on common users preferences, known as user-based CF, which employed users' similarities for the formation of the neighborhood of nearest users. Many improvements of user-based CF have been suggested, e.g., (Breese et al., 1998). Moreover, item-based CF (Sarwar et al., 2001; karypis, 2001) is based on the items' similarities for a neighborhood generation of nearest items. Many model-based algorithms have been also developed (Breese et al., 1998). Sarwar et al. (2000) proposed a Matrix Factorization model based on Singular Value Decomposition (SVD) for generating rating predictions. Koren (2008), who is member of the winning team in the Netflix prize, proposed SVD++ method, which adds in the plain SVD also information taken from user/item bias and other implicit feedback. As the Netflix prize competition has demonstrated, matrix factorization models are superior to classic nearest-neighbor techniques (Koren, 2008).

There are several methods (Goldbeck, 2005; Jamali and Ester, 2010; Vasuki et al., 2010; Ma et al., 2009, 2011; Yang et al., 2011), that combine information from unipartite and bipartite graphs, focusing in the rating prediction (i.e. item/group recommendation) problem. For example, TidalTrust (Goldbeck, 2005) and MoleTrust (Massa and Avesani, 2004) combine the rating data of collaborative filtering systems with the link data of trust-based social networks to improve the item recommendation accuracy. In particular, TidalTrust (Goldbeck, 2005) performs a modified breadth first search in the trust network to compute a rating prediction. Furthermore, MoleTrust (Massa and Avesani, 2004) considers paths of friends to a user-defined maximum-depth. Recently, Vasuki et al. (2010) proposed affiliation/group recommendations based on the friendship network among users, and the affiliation/group network between users and groups. In particular, they suggested two models of user-community affinity for the purpose of making affiliation recommendations: one based on graph proximity, and another using latent factors to model users and communities. Ma et al. (2009, 2011) proposed a probabilistic matrix factorization framework for recommender systems, which naturally fuses the users tastes and their trusted friends favors together. In the same direction, Jamali and Ester (2010) proposed also a probabilistic matrix factorization technique with trust propagation for leveraging item recommendations in social networks. They explored a model-based approach for item recommendation in social networks, employing matrix factorization techniques, by incorporating the mechanism of trust propagation into their model. However, they have not theoretically considered the ability of their model to take into account more than two social networks. Finally, recently Yang et al. (2011) proposed a probabilistic framework (Friendship Interest Propagation- FIP) for both handling the link and rating prediction problem. However, in contrast to our work, they control the influence and impact of each social network in their model, through a user-defined parameter, which additionally requires user-controlled parameter tuning.

3 Link and Rating Prediction based on user-item bipartite graph

As described in Section 1, users can form several *implicit* social networks through their daily interactions like co-commenting on people's post, co-rating similarly products, and co-tagging people's photos. These implicit relations, contain edges between two types of entities, such as a user-item bipartite graph. An example can be seen in Figure 1, which presents the ratings of users on items and will be also used in our running example.

The user-item bipartite graph of our running example, can be also presented by a matrix R, where the rating of a user u over an item i is given from the element $R(u, i)$. An example of such a matrix is given in Figure 2, where I_{1-2} are items and U_{1-4} are users. The null cells (no rating) are presented with dash.

Related work in Collaborative Filtering (Herlocker et al., 2002; McLauglin and Herlocker, 2004; Mobasher et al., 2001; Sarwar et al., 2001) has used Pearson correlation or Cosine similarity to compute similarity among users of a bipartite

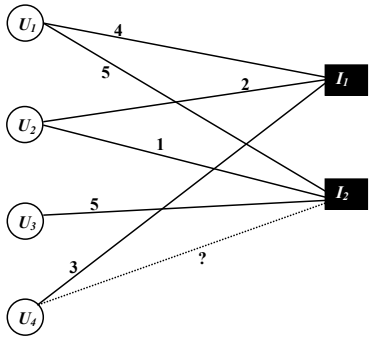

Figure 1 Example of a user-item bipartite network.

	I_1	I_2
U_1	4	5
U_2	2	1
U_3	-	5
U_4	3	-

Figure 2 Running example: User-Item matrix R.

graph. In our implementation, we will use the cosine similarity (Equation 1), which measures the similarity between two users, u and v.

$$\text{sim}(u,v) = \frac{\sum\limits_{\forall i \in I_u \cap I_v} (r_{u,i} \cdot r_{v,i})}{\sqrt{\sum\limits_{\forall i \in I_u} (r_{u,i})^2} \sqrt{\sum\limits_{\forall i \in I_v} (r_{v,i})^2}} \tag{1}$$

Note that I_u, I_v are the sets that contain the rated items by the users u and v, respectively. Therefore, $I_u \cap I_v$ denotes the co-rated items by u, v.

The application of Equation 1 to our running example, constructs the *rating similarity matrix* sim_R, which is depicted in Figure 3.

	U_1	U_2	U_3	U_4
U_1	1	0.908	0.781	0.625
U_2	0.908	1	0.447	0.894
U_3	0.781	0.447	1	0
U_4	0.625	0.894	0	1

Figure 3 sim_R similarity matrix based on rating matrix.

For the link prediction task, in our running example, let's assume that we want to propose new friends to user U_4. As shown in Figure 3, we can rank the similarities

of user U_4 with the other users (i.e. U_1, U_2). Then, we can recommend to U_4 as top friend user U_2.

For the rating prediction task, in our running example, let's assume, that we want to predict the rating of user U_4 on item I_2. We can also take into account the similarities between users (see Figure 3). Then, we can compute the predicted rating of a user u for an item i by using Equation 2:

$$p_{u,i} = \frac{\sum_{v \in U} [\text{sim}(u,v) \cdot r_{v,i}]}{\sum_{v \in U} \text{sim}(u,v)} \tag{2}$$

Note that in Equation 2 only the users v that have rated the item i are involved in the summations.

Based on Equation 2, and by selecting two users as the neighborhood of U_4 (i.e. nearest users U_2 and U_1, the rating prediction of U_4 on item I_2, is equal to 2.646 $[(0.894 * 1 + 0.625 * 5)/(0.894 + 0.625) = 2.646]$.

4 Link and rating prediction based on user-user unipartite graph

As described in Section 1, users can form an *explicit* social network by adding each other as friends. Let $\mathcal{G}$ be a graph with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. Every edge is defined by a specific pair of graph nodes (v_i, v_j), where $v_i, v_j \in \mathcal{V}$. An example of such a graph can be seen in Figure 4, which presents a friendship network and will be used as our running example.

We assume that the graph $\mathcal{G}$ is undirected and un-weighted, thus the graph edges do not have any weights, plus the order of nodes in an edge is not important. Therefore, (v_i, v_j) and (v_j, v_i) denote the same edge on $\mathcal{G}$. Moreover, graph $\mathcal{G}$ can not have multiple edges that connect two nodes, thus if two nodes v_i, v_j are connected with an edge of $\mathcal{E}$, then there can not exist another edge in $\mathcal{E}$ also connecting them. Finally, we assume that there can not be loop edges on $\mathcal{G}$ (i.e. a node can not be connected to itself).

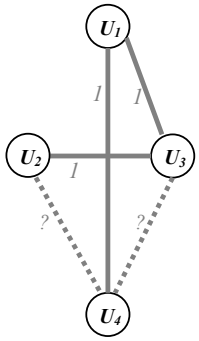

Figure 4 Example of a unipartite friendship Network.

A common graph representation is the *adjacency matrix* A. It is an $n \times n$ matrix, where $n = |\mathcal{V}|$ is the number of nodes in $\mathcal{G}$. Therefore, it has n rows and n columns labelled by the graph nodes. For an un-weighted non-multiple graph (such as $\mathcal{G}$), the adjacency matrix values are set as follows:

$$A[v_i, v_j] = \begin{cases} 1, & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0, & \text{if } (v_i, v_j) \notin \mathcal{E} \end{cases}$$

Following all previous assumptions and definitions, the adjacency matrix of an undirected and un-weighted graph such as $\mathcal{G}$, is a symmetric matrix with values 1 and 0, if two nodes are neighbors or not, respectively. In addition, as there are not any loop edges, the main matrix diagonal have zero values. The adjacency matrix of our running example is depicted in Figure 1.

	U_1	U_2	U_3	U_4
U_1	0	0	1	1
U_2	0	0	1	0
U_3	1	1	0	0
U_4	1	0	0	0

Table 1 Running example: User-User Adjacency matrix A.

For the link prediction task, let's assume in our running example, that we want to recommend new friends to user U_4. As discussed in Section 2, there is a variety of local similarity measures (Liben-Nowell and Kleinberg, 2003) (i.e. Adamic/Adar index, Jaccard Coefficient, Common Neighbors index etc.) for analyzing the "proximity" of nodes in a network. We use an extension of the Jaccard Coefficient (i.e. Tanimoto coefficient (Tanimoto, 1957)), which is a cosine similarity metric that measures the degree of overlap between node vectors, as shown in the following Equation:

$$sim(v_i, v_j) = \begin{cases} 0, & \text{if } (v_i, v_j) \notin \mathcal{E} \\ \frac{1}{deg(v_i)+deg(v_j)-1}, & \text{otherwise} \end{cases} \tag{3}$$

where $deg(v_i)$ and $deg(v_j)$ are the degrees of nodes v_i and v_j, respectively.

Collecting all similarity values between the nodes of a graph $\mathcal{G}$, we construct the *basic node similarity matrix* sim_A of $\mathcal{G}$, which is an $n \times n$ matrix having n rows and n columns labeled by the graph nodes. In our running example, sim_A is depicted in Figure 5.

As shown, we can not infer new interactions for user U_4, since there are no connections between the corresponding users. Notice also that the similarity values between all non-adjacent nodes are zero. For instance, in our running example, the similarity value between nodes U_4 and U_3 is zero, because they do not share any edge. However, users U_4 and U_3 have both user U_1 as a common friend, and thus they could be related in some way.

By using a transitive similarity we can efficiently solve this problem. We can define a transitive node similarity, between two nodes v_i and v_j, denoted as extended

	U_1	U_2	U_3	U_4
U_1	0	0	0.333	0.5
U_2	0	0	0.5	0
U_3	0.333	0.5	0	0
U_4	0.5	0	0	0

Figure 5 sim_A Similarity Matrix based on Adjacency Matrix.

similarity. Extended similarity is calculated by the product of the basic similarities between the nodes of the shortest path from v_i to v_j (if there are more than 1 shortest paths it is averaged).

In our running example, the extended similarity between nodes U_4 and U_3 equals:

$$sim(U_4, U_3) = sim(U_4, U_1) \cdot sim(U_1, U_3) = \frac{1}{2} \cdot \frac{1}{3} = \frac{1}{6} = 0.167$$

Moreover, the extended similarity between nodes U_4 and U_2 equals:

$$sim(U_4, U_2) = sim(U_4, U_1) \cdot sim(U_1, U_3) \cdot sim(U_3, U_2) =$$

$$= \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{12} = 0.083$$

Collecting all the extended similarity values between the nodes of a graph $\mathcal{G}$, we can update sim_A with the new extended similarity values. In our running example, the updated sim_A is depicted in Figure 6. As shown in Figure 6, we can recommend to user U_4 as top friend user U_3.

	U_1	U_2	U_3	U_4
U_1	0	0.167	0.33	0.5
U_2	0.167	0	0.5	0.083
U_3	0.33	0.5	0	0.167
U_4	0.5	0.083	0.167	0

Figure 6 Updated sim_A Similarity Matrix with the extended similarities.

For the rating prediction task, we can predict the rating of a user on an item based on the above similarity matrix by taking also into account the ratings of the target user's nearest neighbors using Equation 2. Based on Equation 2, and by selecting two users as the neighborhood of U_4 (i.e. nearest users are U_1 and U_3, the rating prediction of U_4 on item I_2, is equal to 5 $[(0.5 * 5 + 0.167 * 5)/(0.5 + 0.167) = 5]$.

5 The Social-Union Framework

In the previous two Sections, we have shown that based on different explicit or implicit social networks (i.e. user-user unipartite network or user-item bipartite

network) we can infer different conclusions. For example in our running example, for the link prediction task, based on the user-item bipartite network in Section 3, we recommend to U_4 as top friend user U_2. In contrast, based on the user-user unipartite network in Section 4, we recommend to U_4 as top friend user U_3. The same stands also for the rating prediction task. Based on the user-item bipartite network, we predict the rating of U_4 on item I_2 to be equal to 2.646, whereas based on the user-user unipartite graph we predict a value equal to 5. It is obvious that each different explicit or implicit social network can be informative and contribute a different aspect of reality.

In this Section, we present our approach, Social-Union, which combines multiple similarity matrices from heterogenous unipartite and bipartite graphs, as discussed in sections 4 and 3, respectively. The multi-modal graph expressing (i) friendship among users and (ii) user ratings on items, can be seen in Figure 7.

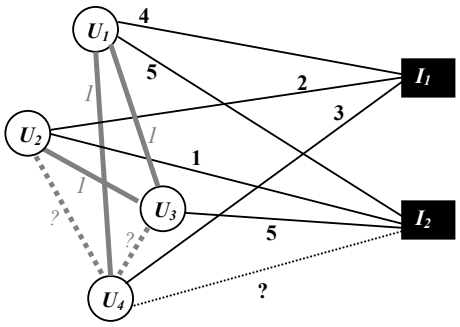

Figure 7 Example of a multi-modal Network.

In our running example, both similarity matrices, i.e. similarity matrix sim_R which is based on the user-item network, and similarity matrix sim_A, which is based on friendship network, contain valuable information. There are several basic aggregation functions (i.e. Min, Max, Average etc.) that could be used to combine the values of the two similarity matrices. In this paper, to compute the similarity between two users u and v, we linearly combine sim_A and sim_R matrices into a single one similarity matrix, as shown by Equation 4:

$$\text{sim}(u, v) = (1 - a_u) \cdot \text{sim}_A + a_u \cdot \text{sim}_R \tag{4}$$

In Equation 4, a_u takes values between [0,1]. This parameter can be adjusted by the user. When a_u takes values greater than 0.5, then the similarity values based on ratings matrix have much more impact in the final similarity values than the similarity values based on adjacency matrix. When a_u becomes zero, the final similarity values are exactly the similarity values based on adjacency matrix only. When a_u becomes one, the final similarity values are exactly the similarity values based on ratings matrix only. Notice that we have also tested experimentally other simple aggregation functions such as Min(sim_A,sim_R), Max(sim_A,sim_R), and Avg(sim_A,sim_R), which do not incorporate parameter a_u.

In several cases the distribution of the similarity values in the interval [0,1] between sim_A and sim_R differ significantly. For example, consider the case that the most similarity values in A are normally distributed between 0 and 0.3, whereas the most similarity values in R are normally distributed between 0.4 and 0.7. Then, it is unfair to take a simple weighted average of them using Equation 4, because the similarity values of A will always be dominated by those of R, and only few small values of the a_u parameter can be chosen for equivalence. In opposition, we want the user to be able to use the a_u parameter freely in the whole range. Therefore, in that case, we use the following transformation procedure for $X = A$ and $X = R$:

- We compute the mean similarity value m_X of the matrix X.
- We compute the standard deviation value s_X of the matrix X.
- We make the following transformations to all similarity values of the X matrix:

$$sim_X(u, v) = \frac{sim_X(u, v) - m_X}{s_X} \tag{5}$$

- Finally, we scale and translate the derived similarity values back in the interval [0,1]:

$$sim_X(u, v) = \frac{sim_X(u, v) - min_X}{max_X - min_X}$$

where min_X, max_X are the minimum and the maximum derived similarity values in matrix X after the transformation of Equation 5, respectively.

5.1 Auto adjustment of the a_u parameter

The adjustment of a_u parameter by the user is definitely useful when the user requires to calibrate manually the similarity calculations. For example, consider the scenario that the final similarity values are used in a clustering application. Then, by varying the a_u parameter, the user can calibrate the number of final derived clusters to the desired. However, there are also applications where the manual adjustment of the a_u parameter does not have a visual impact into their results, thus the user cannot manage the variations and requires an auto-adjustment.

As described in Section 2, there are local and global features of social networks. By Equation 6, we provide an independent automatic adjustment of a_u parameter that takes into account the local (i.e. user's density) and global (i.e. network's density) characteristics of multi-modal graphs:

$$a_u = \frac{dR}{dA + dR} \tag{6}$$

where:

- $dA = \frac{localA}{globalA}$ is the local to global density coefficient of the selected user u into the adjacency matrix A. $localA$ is the local density of the selected user u into the adjacency matrix, i.e. the number of non-zero values in its row divided by the number of users ($deg(u)/n$). $globalA$ is the global density of the adjacency matrix, i.e. the number of non-zero values in the full A matrix divided by the square of number of users ($/n^2$).

- $dR = \frac{localR}{globalR}$ is the local to global density coefficient of the selected user u into the ratings matrix R. $localR$ is the local density of the selected user u into the ratings matrix, i.e. the number of non-zero values in its row (non-zero ratings) divided by the number of items $|I|$. $globalR$ is the global density of the ratings matrix, i.e. the number of non-zero values in the full R matrix divided by the number of users and the number of items ($/(n|I|)$).

The construction and derivation of Formula 6 is simple: The denominator expresses the total weighted density from both R and A data, whereas the nominator expresses the total weighted density only from the R data, as a_u expresses the ratio of the similarity values based on R (see Equation 4).

By substitution of dA, dR in Equation 6 we have:

$$a_u = \frac{localR \cdot globalA}{localA \cdot globalR + localR \cdot globalA} \tag{7}$$

in case we want to express the a_u parameter using only the local and global densities.

5.2 Generalization of Social-Union for more similarity matrices

As already mentioned, a_u corresponds to the ratio of the similarity based on the ratings matrix in Equation 4. Therefore, let us denote it as $a_{u,R}$, i.e. $a_{u,R} = a_u$. Moreover, $(1 - a_u)$ corresponds to the ratio of the similarity based on the adjacency matrix in Equation 4. Therefore, let us denote it as $a_{u,A}$, i.e. $a_{u,A} = 1 - a_u$. By substituting parameter a_u in Equation 4 with Equation 6, the Social-Union similarity becomes:

$$sim(u, v) = \frac{dA}{dA + dR} \cdot sim_A + \frac{dR}{dA + dR} \cdot sim_R$$

or equivalently:

$$sim(u, v) = a_{u,A} \cdot sim_A + a_{u,R} \cdot sim_R \tag{8}$$

Therefore, we have a specific general rule for the auto adjustment of the ratio similarity coefficients: *$a_{u,X}$ is always equal to its corresponding final density coefficient dX divided by the sum of all existing final density coefficients.*

Now, Equation 8 can be generalized for any number of additional similarity matrices as follows:

$$sim(u, v) = a_{u,S_1} \cdot sim_{S_1} + a_{u,S_2} \cdot sim_{S_2} + ... + a_{u,S_k} \cdot sim_{S_k} \tag{9}$$

where:

$$a_{u,S_i} = \frac{dS_i}{dS_1 + dS_2 + ... + dS_k} \tag{10}$$

and

$$dS_i = \frac{localS_i}{globalS_i} \tag{11}$$

and $localS_i, globalS_i$ are the local and global densities for the matrix $S_i, \forall i = 1, 2, ..., k$ as defined previously.

5.3 The Social-Union Algorithm for Link Prediction

Figure 8 depicts the outline of the algorithmic procedure of the proposed methodology. In lines $1-8$, the similarity values of each source network are calculated separately for the selected user u. In lines 9-11, the density values and the coefficients are calculated for the selected user u. In lines 12-13, the unified similarity values are calculated.

Algorithm Social-Union-Link-Prediction
Input
u: the selected user
$SN_i, i = 1, ..., k$: k unipartite or bipartite social networks (including the ratings R)
Output
US: Unified Similarity Matrix

---

01. **for** $i = 1$ **to** k
02. **if** SN_i is unipartite **then**
03. compute the similarity values (sim_{S_i}) of SN_i, using Eq. 3 or any other link prediction similarity measure for user u. (i.e. Adamic/ Adar, Common Neighbors, Random Walk with Restart, Katz etc.)
04. **else if** SN_i is bipartite **then**
05. compute the similarity values (sim_{S_i}) of SN_i, using Eq. 1 or any other user-based collaborative filtering similarity measure (i.e. Pearson Correlation, Jaccard Coefficient etc.) for user u.
06. **end if**
07. apply transformation of Eq. 5 in all similarity values of u.
08. **end for**
09. **for** $i = 1$ **to** k
10. compute the local and global densities $localS_i$, $globalS_i$ and their ratio dS_i using Eq. 11 for the user u.
11. **end for**
12. compute the coefficients a_{u,S_i} using Eq. 10 for the user u.
13. compute final similarity values of user u using Eq. 9 in US
14. **return** US.

Figure 8 Outline of the Social Union Link Prediction Algorithm.

5.4 Using Social-Union results for Rating Prediction

To derive rating predictions for a specific user u we take into account the ratings of the top-m similar users to u, where $m < n$ is a user-defined parameter. This methodology was used also in (Breese et al., 1998; Herlocker et al., 1999, 2002).

More specifically, let $s_1, s_2, ..., s_m$ the corresponding final similarity values of the top-m similar users $u_1, u_2, ..., u_m$ to u (those values have already been calculated with the Social-Union methodology, i.e. $s_i = sim(u, u_i)$). Let also, r_{ij} the corresponding known ratings, and avg_i the average ratings value, of the user u_i

in the ratings matrix R, for $i = 1, 2, ..., m$. Let also avg the average known ratings of the user u in the ratings matrix. Then, the predicted ratings for the user u are defined as follows:

$$\widehat{r}_{u,j} = avg + \frac{\sum_{i=1}^{m} s_i \cdot |r_{ij} - avg_i|}{\sum_{i=1}^{m} s_i} \tag{12}$$

where j is any unrated item by the user u.

Finally, we sort the predicted ratings $\widehat{r}_{u,j}$ of user u and we suggest the top-e items, where e is a desired cardinality value.

5.5 The Social-Union Algorithm for Rating Prediction

Figure 9 depicts the outline of the algorithmic procedure of the proposed framework for the rating prediction task. The input is the Unified Similarity Matrix US computed from the Social-Union-Link-Prediction algorithm. The predicted ratings are calculated and the top items are returned to the user.

Algorithm Social-Union-Rating-Prediction
Input
u: the selected user
US: Unified Similarity Matrix computed from Social-Union-Link-Prediction algorithm (including the ratings R)
Output
E: a set of suggested items

---

1. find the top-m similar users $u_1, ..., u_m$ to u.
2. get corresponding final similarity values $s_1, ..., s_m$ from US.
3. get corresponding known ratings r_{ij} of u_i from R.
4. compute the average ratings of the users u_i and u in R.
5. compute the predicted ratings for the user u using Eq. 12
6. put the top-e items with the highest predicted ratings in E.
7. **return** E.

Figure 9 Outline of the Social Union Rating Prediction Algorithm.

6 Complexity and Implementation Issues

The complexity of the proposed algorithms is strongly depended from (i) the selected similarity measures and algorithms used for processing the corresponding unipartite and bipartite networks, and (ii) the structures that are used for keeping the network data and the corresponding indices in memory.

For example, let us assume the case that: (i) we combine one unipartite (user per user) and one bipartite (user per item) network, (ii) we use the FOAF algorithm (Chen et al., 2009) and the user-based Collaborative Filtering (user-based CF) algorithm (Breese et al., 1998) for processing the unipartite and bipartite networks respectively, and (iii) we use an adjacency list for the representation of structures in memory for all networks (using dynamic memory allocation).

The Social-Union-Link-Prediction Algorithm will have the following time complexity:

- (lines 1-8 loop): (i) a complexity $O(deg(u) * (N + M))$ is required for the FOAF calculations, where $deg(u)$ is the degree of the selected user, N is the total number of users, and M is the total number of edges, in the unipartite network, plus (ii) a complexity $O(N + L)$ is required for the user-based CF calculations, where L is the total number of edges in the bipartite network.

- (lines 9-12): these calculations require a complexity of $O(N + M) + O(N + L)$, as we have to scan the two adjacency lists of the networks. Note that this can be done once (in a preprocessing level) and all density values and coefficients can be stored, for later predictions.

- (line 13): the calculation of the final similarity values require a complexity of $O(N)$.

Putting altogether the total required time complexity for the Social-Union-Link-Prediction Algorithm (for this example) will be: $O(deg(u) * (N + M) + L)$.

The complexity of the Social-Union-Rating-Prediction Algorithm is independent from how many bipartite and unipartite networks are used or their selected measures for similarity calculations, as its input is the unified similarities of the user u (a vector of N similarity values). More specifically, the time complexity of this algorithm is:

- (lines 1-2): the top-m similar users are found after sorting the similarity values in the input vector by using a sorting algorithm (quick-sort) with complexity $O(N \log N)$.

- (line 3-5): these calculations require a complexity of $O((m + 1) * I)$, where I is the total number of items in R.

- (line 6): the top-e recommendation items are found after sorting the predicted ratings by using a sorting algorithm (quick-sort) with complexity $O(I \log I)$ (note that in the worst case the user u may have no rankings, thus all I items remain unranked and take predicted values).

Putting altogether the total required time complexity for the Social-Union-Rating-Prediction Algorithm is: $O(N \log N + I \log I)$ (note that usually m is a small number in comparison with $\log I$). The space complexity is depended only from how many networks we have. For the previous example it is $O(N + M) + O(N + L)$ as an adjacency list representation is used for both networks.

7 Experimental Evaluation

In this Section, in the area of the rating prediction task, we compare experimentally our approach, denoted as Social-Union with 3 other algorithms. Our experiments were performed on a 3 GHz Pentium IV, with 2 GB of memory. All algorithms were implemented in C. To evaluate the examined algorithms, we have generated synthetic data sets and chosen two real data sets from Epinions and Flixster web sites. In particular, we consider the following comparison partners:

- **SVD++**: SVD++ (Koren, 2008) is applied on the bipartite user-items rating network. It combines traditional SVD with implicit information taken from implicit user preferences (i.e which items users rate, regardless of their rating). SVD++ was shown to offer accuracy superior to SVD (Koren, 2008). It models users as latent features, while adding implicit feedback. The exact model is as follows:

$$\widehat{r}_{ui} = b_{ui} + q_i^T \cdot (p_u + N(u)^{-0.5} \sum_{j \in N(u)} y_j), \tag{13}$$

  where b_{ui} is the user/item bias, p_u is a user-factors vector and q_i is an item-factors vector, $N(u)$ denotes the set of items for which user u expressed an implicit preference. Model parameters are determined by minimizing the associated regularized squared error function through stochastic gradient descent with extra parameters γ, λ_5, and λ_6 (for more information see (Koren, 2008)). For Epinions data set, the best values of parameter γ, λ_5, and λ_6 are fixed at 0.006, 0.007 and 0.02, respectively. For the Flixter data set, the best values of parameters are fixed at 0.004, 0.006 and 0.03.

- **FriendTNS**: This is a graph-based algorithm (Symeonidis et al., 2010), which takes into account the friendship network. In particular, FriendTNS defines a similarity measure that captures effectively local and global graph features, introducing transitive node similarity. Notice that for the rating prediction evaluation task, we assign the FriendTNS similarity scores in Equation 12.

- **tKatz**: This is a truncated version of Katz (Vasuki et al., 2010) algorithm that uses information from two networks, i.e. user-item rating network and user-user friendship network. It mainly counts paths between networks from different sources, where the heterogeneity of the two types of links is reduced to a single parameter $\lambda \geq 0$, that controls the ratio of the weight of friendship to the weight of item membership. The Katz (Katz, 1953) measure directly sums over all paths between any pair of nodes, exponentially damped by length, to count short paths more heavily (using parameter β). Truncated Katz (i.e. tKatz) chooses to stop after reaching paths of length ℓ_{max}. The similarity between two nodes v_x and v_y, can be computed by Equation 14:

$$tKatz(v_x, v_y) = \sum_{\ell=1}^{\ell_{max}} \beta^{\ell} \cdot \left| paths^{\ell}_{v_x, v_y} \right|, \tag{14}$$

where $\left|paths^{\ell}_{v_x,v_y}\right|$ is the number of all length-ℓ paths from v_x to v_y. The best values of the parameters of tKatz algorithm, for the Epinions and Flixter data sets, have been learnt after appropriate parameters tuning. This tuning has been made by varying the value of one parameter, while having the other parameters fixed. For Epinions data set, the best performance of tKatz has been reached with values of parameter β, λ, and ℓ_{max} at 0.01, 0.3 and 4, respectively. For the Flixter data set, the best values of parameters are set to 0.05, 0.2 and 3.

Moreover, in the area of the link prediction task, except the User-based CF and FriendTNS algorithms, we also consider the following comparison partner, which is an state-of-the-art algorithm for link prediction:

- **SL-H(HS)** : This is a supervised learning (SL) algorithm (Lu et al., 2010) for link prediction that uses information from multiple networks. It mainly counts paths between networks from different sources. We employ a special version of the SL algorithm that uses hybrid color paths and hierarchical structured regularization(H-HS).

7.1 Real Data Sets and multi-Social Generator

To evaluate the examined algorithms, we have generated synthetic data sets and chosen two real data sets from Epinions and Flixster web sites.

We used the Epinions[1] data set, which is a who-trusts-whom social network. In particular, users of Epinions.com express their Web of Trust, i.e. reviewers whose reviews and ratings they have found to be valuable. It contains 49K users with 487K edges among them, and 140K items with 665K ratings.

Moreover, we use the Flixster[2] data set, which is a social networking service in which users can rate movies and add other users to their friend list creating a social network. It contains 1M users with 7M edges among them, and 49K items with 8.2M ratings.

In contrast to purely random (i.e., Erdos-Renyi) graphs, where the connections among nodes are completely independent random events, our synthetic model ensures dependency among the connections of nodes, by characterizing each node with a m-dimensional vector with each element a randomly selected real number in the interval [-1,1]. This vector represents the initial user profile which will be used for the construction of the friendship and ratings profiles.

For the construction of the friendship network, two nodes are considered to be similar and thus of high probability to connect to each other if they share many close attributes in their initial user profile. Given a network size N and a mean degree k of all nodes, we start with an empty network with N nodes. At each time step, a node with the smallest degree is randomly selected (there is more than one node having the smallest degree). Among all other nodes whose degrees are smaller than k, this selected node will connect to the most similar node with probability $1-p$, while a randomly chosen one with probability p. The parameter $p \in [0,1]$ represents the strength of randomness in generating links, which can be understood as noise or irrationality that exists in almost every real system.

For the construction of the user-item rating network, we follow a similar procedure. In addition, we use the following parameters: (i) the ratings range, (ii)

the mean number of rated items by all users. Notice that each user can rate different items from others and has in his profile a different number of rated items.

Based on the above procedure, we have produced a unipartite user-user (friendship) network and a bipartite user-item network. They contain 5K users with 125K edges among them, and 500 items with 250K ratings in the range $[1, 5]$, and mean number of rated items 50. The parameter p is fixed to 0.2. We calculated several topological properties of the synthetic and real data sets for the friendship network, presented in Figure 10a. As shown, Epinions 49K and Flixter 1M present (i) a large clustering coefficient (LCC) equal to 0.26 and 0.18 respectively, and (ii) a small average shortest path length (ASD) equal to 4.0 and 3.16 respectively. These topological features can be mainly discovered in small-worlds networks. Small-world networks have sub-networks that are characterized by the presence of connections between almost any two nodes within them (i.e.high LLC). Most pairs of nodes are connected by at least one short path (i.e. small ASD). Moreover, we calculated basic statistics for the user-item network of the tested data sets, presented in Figure 10b.

TOPOLOGICAL PROPERTIES OF FRIENDSHIP NETWORKS:
N = total number of nodes
E = total number of edges
ASD = average shortest path distance between node pairs
ADEG = average node degree
LCC = average local clustering coefficient
GD = graph diameter (maximum shortest path distance)
GGS = global graph sparsity (number of zero values in adjacency matrix / N^2)

Data-Set	Type	N	E	ASD	ADEG	LCC	GD	GGS
Epinions 49K	Directed	49288	487183	4.00	19.77	0.26	14	99.96%
Flixter 1M	undirected	1049511	7058819	3.16	13.45	0.18	9	99.97%
Synthetic 5K	undirected	5000	125000	2.69	50	0.01	5	99%

(a)

PROPERTIES OF USER-ITEM BIPARTITE NETWORKS:
N = total number of Nodes (users)
R = total number of Ratings
I = total number of Items
MINR = minimum rating value
MAXR = maximum rating value
AVGR = average rating value
GGS = global graph sparsity (number of zero values in matrix / existing users x items)

Data-Set	N	R	I	MINR	MAXR	AVGR	GGS
Epinions 49K	49288	664824	139738	1	5	3.99	99.98%
Flixter 1M	1049511	8196077	100000	0.5	5	3.58	99.95%
Synthetic 5K	5000	250000	500	1	5	3.00	90%

(b)

Figure 10 Topological properties of (a) friendship and (b) user-item networks.

7.2 Experimental Protocol and Evaluation for the rating prediction task

Our evaluation considers the division of rated items of each target user into two sets: (i) the training set $\mathcal{E}^T$ is treated as known information and, (ii) the probe set $\mathcal{E}^P$ is used for testing and no information in the probe set is allowed to be used for prediction. It is obvious that, $\mathcal{E} = \mathcal{E}^T \cup \mathcal{E}^P$ and $\mathcal{E}^T \cap \mathcal{E}^P = \oslash$. Therefore, for a target user we generate the item recommendations based only on the items in $\mathcal{E}^T$.

Real and synthetic data sets do not have time stamps of the edges. The performance of the algorithms is evaluated by applying double cross-validation (internal and external). Each data set was divided into 10 subsets. Each subset ($\mathcal{E}^P$) was in turn used for performance estimation in the external cross-validation. The 9 remaining subsets ($\mathcal{E}^T$) were used for the internal cross-validation. In particular, we performed an internal 9-fold cross-validation to determine the best values of the algorithms' needed parameters. We chose as values for the parameters those providing the best performance on the internal 9-fold cross-validation. Then, their performance is averaged on the external 10-fold cross-validation. The presented results, based on two-tailed t-test, are statistically significant at the 0.05 level.

For the experimental evaluation of ratings prediction task, we used the Root Mean Squared Error (RMSE), which has the nice property to amplify the larger prediction error i.e. the larger difference between real and predicted rating. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{(u,j)\in E^P}(r_{u,j} - \widehat{r}_{u,j})^2}{|E^P|}} \tag{15}$$

where E^P is the set of all pairs (u, j) in the probe set.

For the experimental evaluation of the item recommendation task, we use the classic precision/recall metrics, which measure how well the system can identify items that the user prefers. For a test user receiving a list of e recommended items (top-e list), precision and recall are defined as follows:(i)**Precision** is the ratio of the number of relevant items in the top-e list (i.e., those in the top-e list that belong in the probe set $\mathcal{E}^P$ of items of the target user) to e. (ii)**Recall** is the ratio of the number of relevant items in the top-e list to the total number of relevant items (all items in the probe set $\mathcal{E}^P$ of the target user).

7.3 Social-Union Sensitivity Analysis for the rating prediction task

In this Section, we test (a) the performance of different aggregation functions (i.e. AggrMin, AggrMax, and AggrAvg) for the combination of similarity matrices, which do not incorporate parameter a and (b) how the performance of Social-Union, user-based CF and FriendTNS are affected, when we apply different controllable density in the friendship and in the user-item rating network, respectively.

In section 5, we presented the definition of our linear combined similarity measure (see Equation 4). In this section, we test simple aggregation functions in order to discover the best precision values that we can attain when we recommend a top item to a user. In particular, we have tested the following possible aggregation functions: (i) AggrAvg which averages the similarity values across the similarity matrices (i.e. parameter a=0.5) (ii) AggrMax, which takes into account the maximum entry across the two similarity matrices and (iii) AggrMin, which takes into account the minimum entry between entries of the two similarity matrices. The aggregation functions' performance can be seen in Figure 11 for the synthetic 5K, Epinions and Flixter data sets, respectively. As shown, the best performance in all data sets is attained by AggrAvg, which indicates that appropriate tuning of parameter a could further leverage the precision performance.

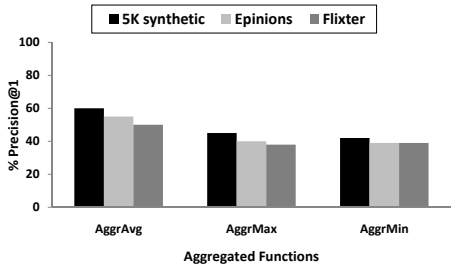

Figure 11 Precision comparison of AggrAvg, AggrMax, AggrMin aggregation functions for the synthetic 5K, Epinions and Flixter data sets.

For the user-item rating network, we have created in our 5K synthetic data set 5 different density cases (i.e 0.2, 0.4, 0.6, 0.8, 1) by changing the fraction of rated items, as shown in Figure 12a. y-axis depicts %*precision*@1, which means the precision we get, when we recommend one item. As expected, as the fraction of rated items increases, precision of Social-Union and SVD++ algorithms increases too. This is reasonable, since every prediction algorithm is expected to give higher accuracy for a denser network. Notice that FriendTNS is stable in all density levels, since it is applied only on the friendship network, whose density is fixed to 0.01.

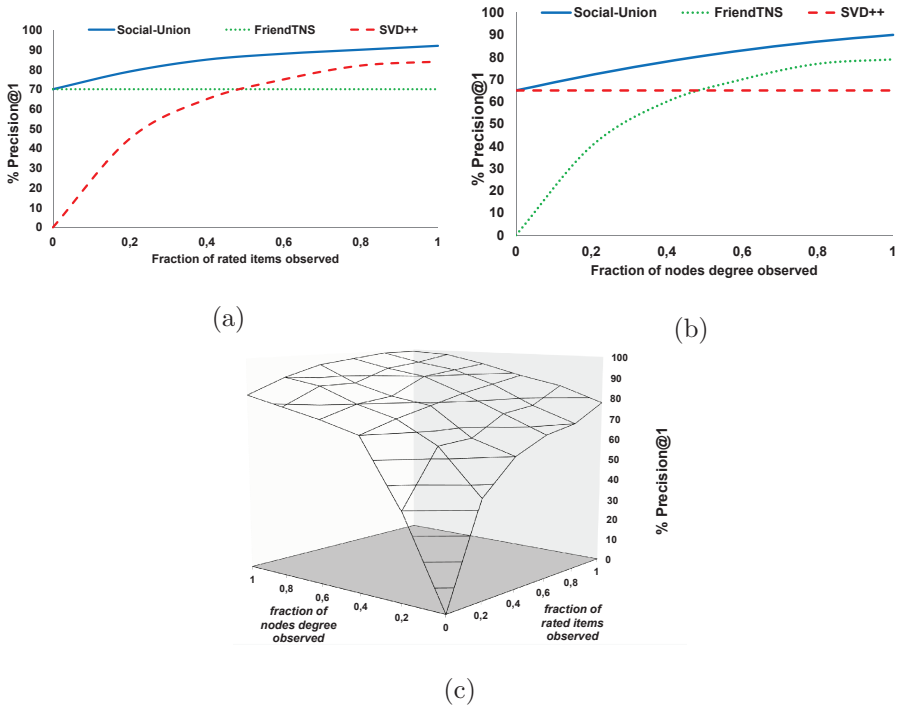

Figure 12 Comparing Social-Union, FriendTNS, and User-based CF with the 5K synthetic data for Precision vs. fraction (a) of rated items (b) of nodes degree and (c) nodes degree and rated items.

Similarly, for the friendship network, we have tested 5 different density cases (i.e 0.2, 0.4, 0.6, 0.8, 1) by changing this time the fraction of edges observed, as shown in Figure 12b. As expected, as the fraction of edges observed increases, precision of Social-Union and FriendTNS algorithms increases too. Notice that SVD++ is stable for all density levels, since it is applied only on the user-rating network, whose density is fixed to 0.1.

Next, we study the performance of Social-Union as we simultaneously increase the fraction of rated items and edges observed in the user-item and friendship networks, respectively. That is, if one parameter's value (i.e. fraction of nodes degree observed) is fixed, we can find the recommendation accuracy of Social-Union varies as we change the other parameter (i.e. fraction of rated items observed). As shown in Figure 12c, Social-Union increases as we increase the fraction of rated items in the user-item network and the fraction of edges observed in the friendship network. We can see there is indeed useful information contained in both friendship and user-item networks topology. This result clearly demonstrates that there is, a significant improvement to be gained by using information from multi-modal social networks.

7.4 Comparison with other Methods for the rating prediction task

In this Section, we proceed with the comparison of Social-Union with the other comparison partners i.e. FriendTNS, SVD++ and tKatz algorithms, in terms of precision and recall. This reveals the robustness of each algorithm in attaining high recall with minimal losses in terms of precision. We examine the top-e ranked item list, which is recommended to a target user, starting from the top item. In this situation, the recall and precision vary as we proceed with the examination of the top-e list.

For the Epinions data set, in Figure 13a we plot a precision vs. recall curve for all four algorithms. As expected, all algorithms' precision falls as e increases. In contrast, as e increases, recall for all algorithms increases as well. Social-Union attains the best results with impressive high precision. The reason is that Social-Union exploits features of both friendship and user-item networks exploiting also local characteristics of the graphs such as node structured density. In contrast, tKatz focuses only on path counts and does not exploit other local network characteristics. Moreover, FriendTNS exploits only information from the friendship network, missing to capture characteristics of the user-item graph. Finally, SVD++ does not take into account information from the friendship network.

For the Flixter data set, in Figure 13b we also plot a precision vs. recall diagram. Social-Union outperforms again the other methods. Notice that the results for Flixter are lower than the results for Epinions for all methods, possibly because the latter has more ratings per user (13.4) and bigger ADEG.

Finally, Table 2 reports the RMSE values of all algorithms on Epinions and Flixter data sets. Again, Social-Union clearly outperforms the other algorithms in terms of RMSE.

7.5 Experimental Protocol and Evaluation for Link Prediction

Our evaluation considers the division of friends of each target user into two sets: (i) the training set $\mathcal{E}^T$ is treated as known information and, (ii) the probe set $\mathcal{E}^P$ is used for testing and no information in the probe set is allowed to be used for

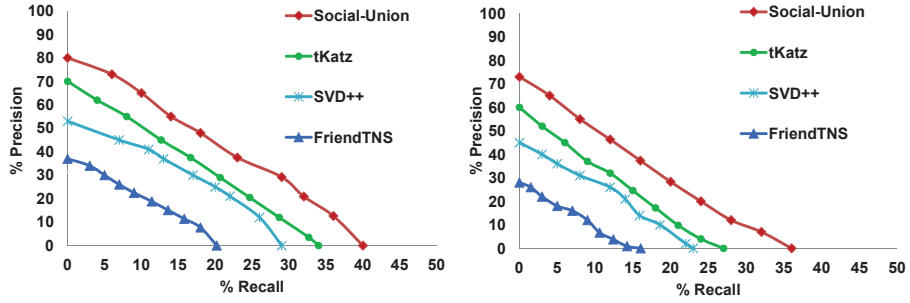

Figure 13 Accuracy performance of algorithms in terms of precision/recall for the (a) Epinions and (b) Flixter data sets.

Table 2 RMSE values for all algorithms on 2 real data sets.

Algorithm	Epinions data set	Flixter data set
Social-Union	**0.765**	**0.812**
tKatz	0.844	0.885
SVD++	0.903	0.941
FriendTNS	1.079	1.121

prediction. It is obvious that, $\mathcal{E} = \mathcal{E}^T \cup \mathcal{E}^P$ and $\mathcal{E}^T \cap \mathcal{E}^P = \oslash$. Therefore, for a target user we generate the recommendations based only on the friends in $\mathcal{E}^T$.

We follow a similar evaluation procedure as described in Section 7.2. We again use the same metrics such as **precision** and **recall**.

7.6 Social-Union Sensitivity Analysis for the link prediction task

In this Section, we compare the performance of (a) different aggregation functions and (b) Social-Union with the other two comparison partners (i.e. SVD++ and Friend-TNS), when we apply different controllable density in the friendship and in the user-item network, respectively.

Similar to the procedure followed before for the rating prediction task, we test the following basic aggregation functions: (i) AggrAvg (i.e. parameter a=0.5) (ii) AggrMax (iii) AggrMin. The aggregation functions' performance can be seen in Figure 14 for the synthetic 5K, Epinions and Flixter data sets, respectively. As shown, we reconfirm the same results with those we had in the rating prediction task, which indicate that appropriate tuning of parameter a could further leverage the precision performance.

For the user-item network, we have created in our 5K synthetic data set 5 different density cases (i.e 0.2, 0.4, 0.6, 0.8, 1) by changing the fraction of rated items, as shown in Figure 15a. As expected, as the fraction of rated items increases, precision of Social-Union and SVD++ algorithms increases too. This is reasonable, since every prediction algorithm is expected to give higher accuracy for a denser

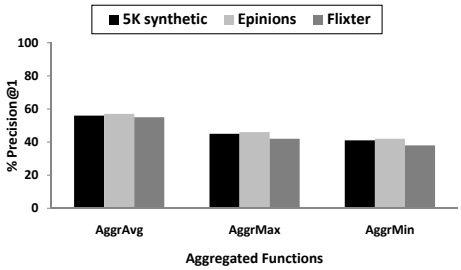

Figure 14 Precision comparison of AggrAvg, AggrMax, AggrMin aggregation functions for the synthetic 5K, Epinions and Flixter data sets.

network. Notice that FriendTNS is stable in all density levels, since it is applied only on the friendship network, whose density is fixed to 0.01.

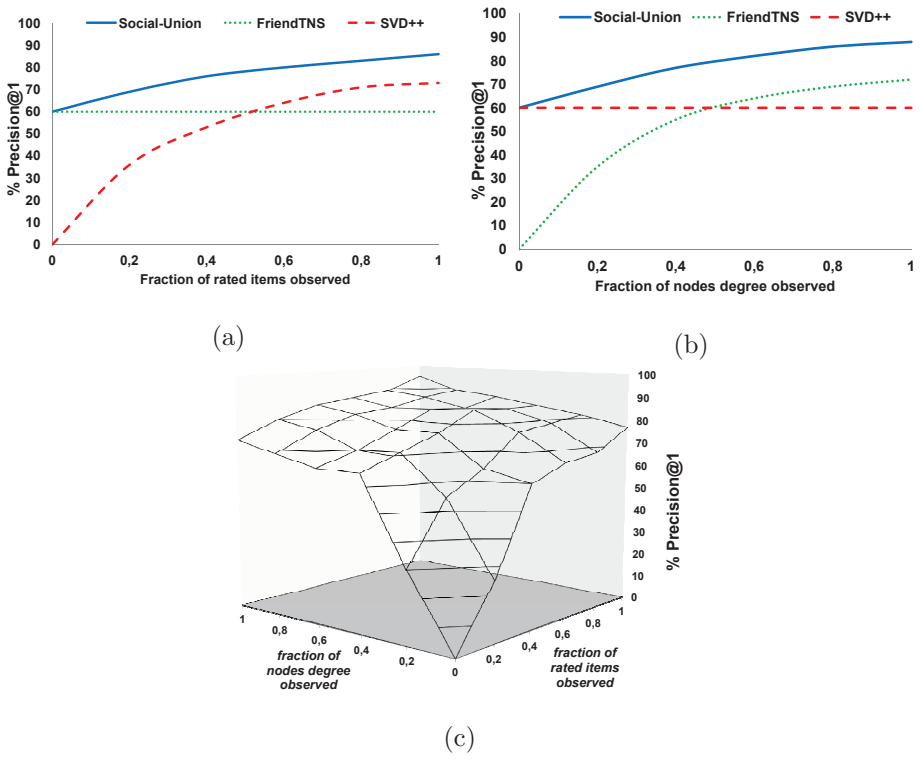

Figure 15 Comparing Social-Union, FriendTNS, and User-based CF with the 5K synthetic data for Precision vs. fraction (a) of rated items (b) of nodes degree and (c) nodes degree and rated items.

Similarly, for the friendship network, we have tested 5 different density cases (i.e 0.2, 0.4, 0.6, 0.8, 1) by changing this time the fraction of edges observed, as shown in Figure 15b. As expected, as the fraction of edges observed increases, precision of

Social-Union and FriendTNS algorithms increases too. Notice that SVD++ is stable for all density levels, since it is applied only on the user-rating network, whose density is fixed to 0.5.

Next, we study the performance of Social-Union as we simultaneously increase the fraction of rated items and edges observed in the user-item and friendship networks, respectively. As shown in Figure 15c, Social-Union increases as we increase the fraction of rated items in the user-item network and the fraction of edges observed in the friendship network. We can see there is indeed useful information contained in both friendship and user-item networks topology.

7.7 Comparison with other Methods for the link prediction task

In this Section, we proceed with the comparison of Social-Union with FriendTNS, SVD++ and SL-H(HS) algorithms, in terms of precision and recall. This reveals the robustness of each algorithm in attaining high recall with minimal losses in terms of precision. We examine the top-k ranked list, which is recommended to a target user, starting from the top friend. In this situation, the recall and precision vary as we proceed with the examination of the top-k list.

For the Epinions data set, in Figure 16a we plot a precision vs. recall curve for all four algorithms. As expected, all algorithms' precision falls as k increases. In contrast, as k increases, recall for all algorithms increases as well. Social-Union attains the best results with impressive high precision. The reason is that Social-Union exploits features of both friendship and user-item networks exploiting also local characteristics of the graphs such as node structured density. In contrast, SL-H(HS) focuses only on path counts and does not exploit other local network characteristics. Moreover, FriendTNS exploits only information from the friendship network, missing to capture characteristics of the user-item graph. Finally, SVD++ does not take into account information from the friendship network. For the Flixter data set, in Figure 16b we also plot a precision vs. recall diagram. Social-Union outperforms again the other methods.

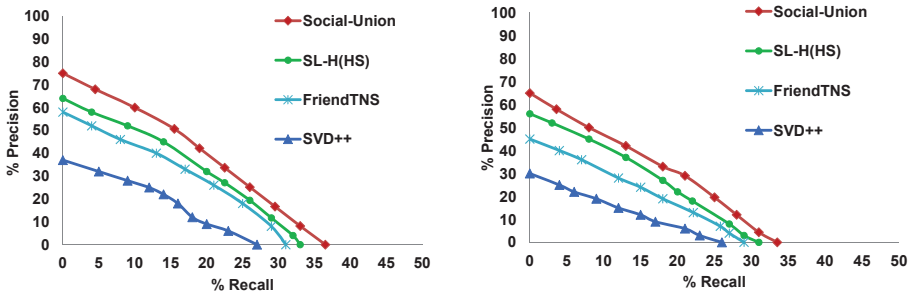

Figure 16 Accuracy performance of Social-Union in terms of precision/recall for the (a) Epinions and (b) Flixter data sets.

8 Conclusions

In this paper, we introduced a generalized framework that exploits multi-modal social networks to provide link and rating prediction in MSNs. We performed extensive experimental comparison of our method Social-Union, against existing well-known link and rating prediction algorithms, using a synthetic and two real data sets (Epinions and Flixter). We have experimentally shown that our Social-Union framework yields to more accurate link and rating predictions. In the future, except unipartite and bipartite graphs, we will extend this framework by incorporating also other higher-order implicit social networks such as tri-partite graphs (e.g. social tagging systems with users, items and tags).

References

Adamic, L. and Adar, E. How to search a social network. *Social Networks*, 27(3):187–203, 2005.

Breese, J., Heckerman, D. and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Confererence on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

Chen, J., Geyer, W., Dugan, C., Muller, M. and Guy, I. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of 27th International Conference on Human Factors in Computing Systems (CHI)*, pages 201–210, 2009.

Golbeck, J. Personalizing applications through integration of inferred trust values in semantic web-based social networks. In *Proceedings of Semantic Network Analysis Workshop at the 4th International Semantic Web Conference*, 2005.

Guy, I., Ronen, I. and Wilcox, E. Do you know?: recommending people to invite into your social network. In *Proceedings of 13th International Conference on Intelligent User Interfaces (IUI)*, pages 77–86, 2009.

Herlocker, J., Konstan, J., Borchers, A. and Riedl, J. An algorithmic framework for performing collaborative filtering. In *Proceedings of ACM SIGIR Conference*, pages 230–237, 1999.

Herlocker, J., Konstan, J. and Riedl, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):287–310, 2002.

Jamali, M. and Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142, 2010.

Jeh, G. and Widom, J. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, 2002.

Karypis, G. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the ACM CIKM Conference*, pages 247–254, 2001.

Katz, L. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

Liben-Nowell, D. and Kleinberg, J. The link prediction problem for social networks. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, 2003.

Lu, Z., Savas, B., Tang, W. and Dhillon, I. Supervised link prediction using multiple sources. In *Proceedings of the ICDM conference on Data Mining*, pages 923–928, 2010.

Ma, H., King, I. and Lyu, M.R. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210,2009.

Ma, H., King, I. and Lyu, M.R. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2 (3):29–48,2011.

Massa, P. and Avesani, P. Trust-aware collaborative filtering for recommender systems. In *Proceedings of Federated Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, pages 492–508, 2004.

McLauglin, R. and Herlocher, J. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the ACM SIGIR Conference*, pages 329–336, 2004.

Mobasher, B., Dai, H., Luo, T. and Nakagawa, M. Improving the effectiveness of collaborative filtering on anonymous web usage data. In *Proceedings of the Workshop on Intelligent Techniques for Web Personalization (ITWP)*, pages 53–60, 2001.

Pan, J., Yang, H., Faloutsos, C. and Duygulu, P. Automatic multimedia cross-modal correlation discovery. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 653–658, Seattle, WA, 2004.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. Grouplens: An open architecture for collaborative filtering on netnews. In *Proceedings of the Computer Supported Collaborative Work Conference*, pages 175–186, 1994.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. Application of dimensionality reduction in recommender system-a case study. In *Proceedings of the ACM WebKDD Workshop*, 2000.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the WWW Conference*, pages 285–295, 2001.

Symeonidis, P., Tiakas, E. and Manolopoulos, Y. Transitive node similarity for link prediction in social networks with positive and negative links. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 183–190, 2010.

Symeonidis, P., Tiakas, E. and Manolopoulos, Y. Product recommendation and rating prediction based on multi-modal social networks. In *Proceedings of the 5th ACM Conference in Recommender Systems (RecSys'2011)*, pages 61–68, Chicago, IL, 2011.

Tanimoto, T. Tanimoto coefficient. *IBM Internal Technical Report*, 1957.

Vasuki, V., Natarajan, N., Lu, Z. and Dhillon, I. Affiliation recommendation using auxiliary networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 103–110, 2010.

Yang, S., Long, B., Smola, A., Sadagopan, N., Zheng, Z. and Zha, H. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th ACM conference on World Wide Web*, 2011.

Yuan, Q., Zhao, S., Chen, L., Liu, Y., Ding, S., Zhang, X. and Zheng, W. Augmenting collaborative recommender by fusing explicit social relationships. In *Proceedings of the Recommender Systems & the Social Web workshop, RECSYS'2009*, 2009.