Product Recommendation and Rating Prediction based on Multi-modal Social Networks

Panagiotis Symeonidis
Department of Informatics
Aristotle University
Thessaloniki, 54124, Greece
symeon@csd.auth.gr

Eleftherios Tiakas
Department of Informatics
Aristotle University
Thessaloniki, 54124, Greece
tiakas@csd.auth.gr

Yannis Manolopoulos
Department of Informatics
Aristotle University
Thessaloniki, 54124, Greece
manolopo@csd.auth.gr

ABSTRACT

Online Social Rating Networks (SRNs) such as Epinions and Flixter, allow users to form several *implicit* social networks, through their daily interactions like co-commenting on the same products, or similarly co-rating products. The majority of earlier work in Rating Prediction and Recommendation of products (e.g. Collaborative Filtering) mainly takes into account ratings of users on products. However, in SRNs users can also built their *explicit* social network by adding each other as friends. In this paper, we propose Social-Union, a method which combines similarity matrices derived from heterogeneous (unipartite and bipartite) explicit or implicit SRNs. Moreover, we propose an effective weighting strategy of SRNs 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 rating prediction and product recommendation algorithms, using synthetic and two real data sets (Epinions and Flixter). Our experimental results show that our Social-Union algorithm is more effective in predicting rating and recommending products in SRNs.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Algorithms, Performance

1. INTRODUCTION

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. A similar situation stands for authors who co-authored a research paper, but also have co-cited the same papers or attended the same conferences. These implicit relations, contain edges between two types of entities, such as user-item bipartite graphs.

There is extensive research [3, 8, 19] in rating prediction and item recommendation from two or more social networks. Recently, Vasuki et al. [19] 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. [20] proposed a linear method to integrate explicit social relationships into Collaborative Filtering methods. However, they did not use a weighting strategy of SRNs influence based on their structured data density.

In this paper, we propose Social-Union, a method which combines multiple similarity matrices derived from heterogeneous explicit or implicit social networks. Social-Union 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 multimodal 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.

RecSys'11, October 23–27, 2011, Chicago, Illinois, USA.

The rest of this paper is organized as follows. Section 2 summarizes the related work, whereas Sections 3 and 4 present how rating prediction is performed from bipartite and unipartite graphs, respectively. Our proposed algorithm, which is based on heterogeneous social networks and several extensions are described in Section 5. Experimental results are given in Section 6. Finally, Section 7 concludes this paper.

2. RELATED WORK

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 [16] 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., [1, 5]. Moreover, item-based CF [17, 9] is based on the items' similarities for a neighborhood generation of nearest items. Most recent work followed the two aforementioned directions (i.e., user-based and item-based). Herlocker et al. [6] weight similarities by the number of common ratings between users/items. Deshpande and Karypis [2] apply item-based CF combined with conditional-based probability similarity and Cosine Similarity.

There are several methods [3, 8, 19], that combine information from unipartite and bipartite graphs, focusing in the rating prediction (i.e. item/group recommendation) problem. For example, TidalTrust [3] and MoleTrust [13] 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 [3] performs a modified breadth first search in the trust network to compute a rating prediction. Furthermore, MoleTrust [13] considers paths of friends to a user-defined maximum-depth. Recently, Vasuki et al. [19] 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. Moreover, Jamali and Ester [8] proposed a 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. Moreover, they control the influence and impact of each social network in their model, through a user-defined parameter λ_T, which additionally requires user-controlled parameter tuning. Moreover, Li et al. [11] proposed a novel model, called *AffRank*, that utilizes an array of six features (product community size, member connectivity, social context, affinity rank history, evolution distance, and average rating) to predict the future rank of products according to their affinities. Finally, He and Chu [4] developed a probabilistic model to make item recommendations based on information in social networks, including user preferences, item's general acceptance and influence from social friends.

3. RATING PREDICTION BASED ON USER-ITEM BIPARTITE GRAPH

As described in Section 1, users can form several *implicit* social rating networks through their daily interactions

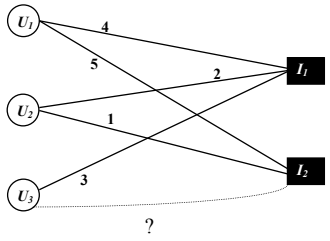


Figure 1: User-Item bipartite network example.

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 Table 1, where I_{1-2} are items and U_{1-3} are users. The null cells (no rating) are presented with dash.

	I_1	I_2
U_1	4	5
U_2	2	1
U_3	3	-

Table 1: Running example: User-Item matrix R.

Related work in Collaborative Filtering [6, 14, 15, 17] has used Pearson correlation or Cosine similarity to compute similarity among users of a bipartite graph. In our implementation, we will use the cosine similarity (Equation 1), which measures the similarity between two users, u and v, where $r_{x,i} = R(x, i)$.

$$\text{sim}(u, v) = \frac{\sum_{\forall i \in I} (r_{u,i} \cdot r_{v,i})}{\sqrt{\sum_{\forall i \in I} (r_{u,i})^2} \sqrt{\sum_{\forall i \in I} (r_{v,i})^2}} \quad (1)$$

The application of Equation 1 to our running example, constructs the *rating similarity matrix* sim_R, which is depicted in Table 2.

	U_1	U_2	U_3
U_1	1	0.908	0.625
U_2	0.908	1	0.894
U_3	0.625	0.894	1

Table 2: sim_R similarity matrix based on rating matrix.

Let's assume in our running example, that we want to predict the rating of user U_3 on item I_2. Based on Table 1, someone could suggest the predicted rating of U_3 on item I_2 to be equal to 3, by simply averaging the ratings of users U_1 and U_2 on item I_2 [e.g. $(5+1)/2 = 3$], since U_3 has equal distance from users U_1 and U_2 based on their rating on I_1. However, if we take into account also the cosine similarity between users (see Table 2), then we compute the predicted rating of a user for an item by using Equation 2:

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

Based on Equation 2, in our running example, the rating prediction of U_3 on item I_2, is equal to 3.709 $[(0.624 * 5 + 0.894 * 1)/(0.624 + 0.894) = 3.709]$.

4. 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 2, 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 $\mathcal{G}$ cannot 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 cannot exist another edge in $\mathcal{E}$ also connecting them. Finally, we assume that there cannot be loop edges on $\mathcal{G}$ (i.e. a node cannot be connected to itself).

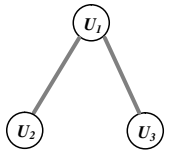


Figure 2: Unipartite friendship Network example.

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 $\mathcal{G}$. The adjacency matrix of our running example is depicted in Table 3.

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

Table 3: Running example: User-User adjacency matrix A.

There is a variety of similarity measures [12] (i.e. Adamic and Adar index, FriendTNS, Jaccard Coefficient, Common Neighbors index, Random Walk with Restart (RWR) etc.) for analyzing the "proximity" of nodes in a network. We use the FriendTNS [18] similarity measure, because it is experimentally [18] shown that outperforms the aforementioned measures. However, any of the aforementioned similarity measures can be alternatively applied in our approach. The intuition behind FriendTNS is that if the shortest path between two nodes goes through a lot of high degree nodes, the two nodes are less likely to become friends. This similarity measure is defined as follows:

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

where $deg(v_i)$ and $deg(v_j)$ are the degrees of nodes v_i and v_j, respectively. For non-adjacent nodes v_i, v_j, we multiply the similarity values between the intermediate nodes of the shortest path between v_i and v_j.

Collecting all similarity values between the nodes of a graph $\mathcal{G}$, we construct the *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 Table 4. As shown, we can infer new interactions for user U_3 (e.g. $sim(U_3, U_2) = sim(U_3, U_1) \cdot sim(U_1, U_2) = \frac{1}{1+2-1} \cdot \frac{1}{2+1-1} = 0.25$).

	U_1	U_2	U_3
U_1	0	0.5	0.5
U_2	0.5	0	0.25
U_3	0.5	0.25	0

Table 4: sim_A Similarity Matrix based on Adjacency Matrix.

Again let's assume in our running example that we want to predict the rating of user U_3 on item I_2. Based on Equation 2, in our running example, the rating prediction of U_3 on item I_2, is equal to 3.66 $[0.5*5+0.25*1)/(0.5+0.25) = 3.66]$.

5. SOCIAL-UNION: RATING PREDICTION BASED ON MULTI-MODAL GRAPHS

In this section, we present our approach, Social-Union, which combines multiple similarity matrices from heterogeneous 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 3.

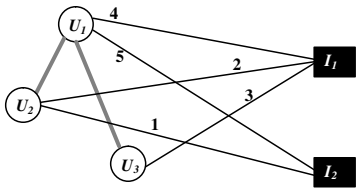


Figure 3: Example of a multi-modal Network.

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. To measure the similarity between two users u and v of a multi-modal social network, we combine sim_A and sim_R matrices into a single one similarity matrix, as shown by Equation 4:

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

In Equation 4, a takes values between [0,1]. This parameter can be adjusted by the user. When a 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 becomes zero, the final similarity values are exactly the similarity values based on adjacency matrix only. When a becomes one, the final similarity values are exactly the similarity values based on ratings matrix only.

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 parameter can be chosen for equivalence. In opposition, we want the user to be able to use the a 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].

5.1 Auto adjustment of the a parameter

The adjustment of a 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 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 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 parameter that takes into account the local (i.e. user's density) and global (i.e. network's density) characteristics of multi-modal graphs:

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

where:

- $dA = \frac{localA}{globalA}$ is the local to global density coefficient of the selected user 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 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 expresses the ratio of the similarity values based on R (see Equation 4).

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

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

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

5.2 Generalization of Social-Union for more similarity matrices

As already mentioned, a corresponds to the ratio of the similarity based on the ratings matrix in Equation 4. Therefore, let us denote it as a_R, i.e. $a_R = a$. Moreover, $(1 - a)$ corresponds to the ratio of the similarity based on the adjacency matrix in Equation 4. Therefore, let us denote it as a_A, i.e. $a_A = 1 - a$. By substituting parameter a 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:

$$sim(u,v) = a_A \cdot sim_A + a_R \cdot sim_R \tag{8}$$

Therefore, we have a specific general rule for the auto adjustment of the ratio similarity coefficients: *a_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:

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

where:

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

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 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 [1, 5, 6].

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:

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

where j is any unrated item by the user u.

If some r_{ij} are not defined into the R matrix (i.e. the user u_i has not rated the item j), then the corresponding terms into the previous summations of equation 12 are deleted.

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.

5.4 The Social-Union Algorithm

Figure 4 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. Finally, in lines 14-20, the predicted ratings are calculated and the top items are returned to the user.

Algorithm Social-Union
Input
u: the selected user
$SN_i, i = 1, ..., k$: k unipartite or bipartite social networks (including the ratings R)
Output
E: a set of suggested items

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 index 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 the 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_{S_i} using Eq. 10 for the user u.
13. compute final similarity values of user u using Eq. 9 in US
14. find the top-m similar users $u_1, ..., u_m$ to u.
15. get the corresponding final similarity values $s_1, ..., s_m$ from US.
16. get the corresponding known ratings r_{ij} of u_i from R.
17. compute the average ratings of the users u_i and u in R.
18. compute the predicted ratings for the user u using Eq. 12
19. put the top-e items with the highest predicted ratings in E.
20. **return** E.

Figure 4: Outline of the proposed methodology.

6. EXPERIMENTAL EVALUATION

In this Section, 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:

- **user-based CF**: User-based CF is applied only on the bipartite user-items network. We use in our comparison an improved version [6] of the well-known user-based collaborative filtering algorithm, that weights similarities by the number of common ratings among users. This variation of user-based CF weights the similarity sim between two users with a parameter γ, as follows: $\frac{\max(c,\gamma)}{\gamma} \cdot sim$, where c is the number of co-rated items. The best value of parameter γ, is fixed at 5 and 3 for the Epinions and the Flixter data set, respectively.

- **FriendTNS**: This is a graph-based algorithm [18], which takes into account only the friendship network. In particular, FriendTNS defines a similarity measure that captures effectively local and global graph features, introducing transitive node similarity.

- **tKatz**: This is a truncated version of Katz [19] 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 [10] 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 13:

$$tKatz(v_x, v_y) = \sum_{\ell=1}^{\ell_{max}} \beta^{\ell} \cdot \left| paths^{\ell}_{v_x, v_y} \right|, \quad (13)$$

where $\left| paths^{\ell}_{v_x, v_y} \right|$ is the number of all length-ℓ paths from v_x to v_y.

For Epinions data set, the best values of parameter β, λ, and ℓ_{max} are fixed at 0.01, 0.3 and 4, respectively. For the Flixter data set, the best values of parameters are fixed at 0.05, 0.2 and 3.

6.1 Real Data Sets

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.

[1] http://www.trustlet.org/wiki/Downloaded_Epinions_dataset

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.

6.2 multi-Social Generator

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 5a. 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 subnetworks 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 5b.

6.3 Experimental Protocol and Evaluation

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 $\mathcal{E}^T$.

[2]http://www.cs.sfu.ca/ sja25/personal/datasets/

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 5: Topological properties of (a) friendship and (b) user-item networks.

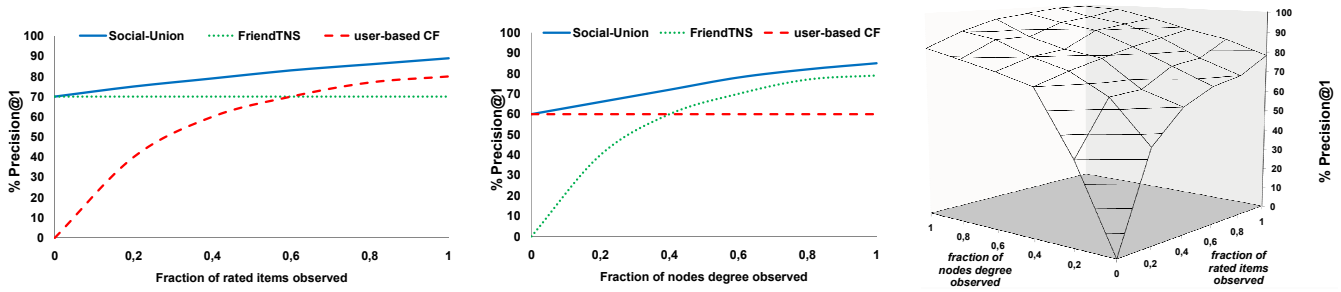
Real and synthetic data sets do not have time stamps of the edges. The performance of the algorithms is evaluated by applying double cross-validation (internal and external). Each data set was divided into 10 subsets. Each subset ($\mathcal{E}^P$) was in turn used for performance estimation in the external 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 [7] to amplify the larger prediction error i.e. the larger difference between real and predicted rating. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{(u,j)\in E^P}(r_{u,j} - \widehat{r}_{u,j})}{|E^P|}} \qquad (14)$$

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 [7], 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: **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. **Recall** is the ratio of the number of relevant items in the top-e list



(a) (b) (c)

Figure 6: 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.

to the total number of relevant items (all items in the probe set $\mathcal{E}^P$ of the target user).

6.4 Social-Union Sensitivity Analysis

In this Section, we test 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.

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 6a. 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 user-based CF 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.

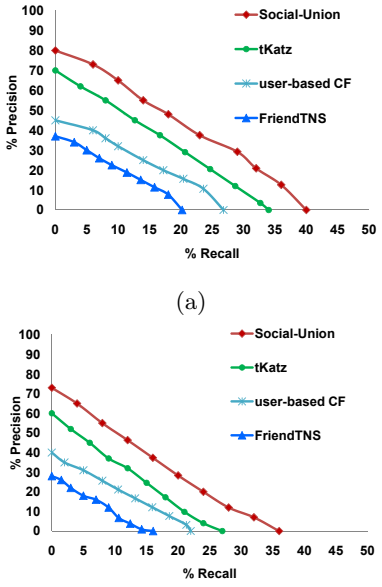
Similarly, for the friendship network, we have tested 5 different density cases (i.e 0.2, 0.4, 0.6, 0.8, 1) by changing this time the fraction of edges observed, as shown in Figure 6b. As expected, as the fraction of edges observed increases, precision of Social-Union and FriendTNS algorithms increases too. Notice that user-based CF 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. As shown in Figure 6c, 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.

6.5 Comparison with other Methods

In this Section, we proceed with the comparison of Social-Union with the other comparison partners i.e. FriendTNS, user-based CF 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 7a 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, user-based CF does not take into account information from the friendship network.



(a)

(b)

Figure 7: Accuracy performance of algorithms in terms of precision/recall for the (a) Epinions and (b) Flixter data sets.

For the Flixter data set, in Figure 7b 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 be-

cause the latter has more ratings per user (13.4) and bigger ADEG.

Finally, Table 5 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.

Table 5: 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
user-based CF	1.013	1.054
FriendTNS	1.079	1.121

7. CONCLUSIONS

In this paper, we introduced a generalized framework that exploits multi-modal social networks to provide item recommendations in SRNs. We performed extensive experimental comparison of our method Social-Union, against existing well-known item recommendation algorithms, using a synthetic and two real data sets (Epinions and Flixter). We have experimentally shown that our Social-Union algorithm yields to more accurate item recommendations. In the future, except item recommendations, we indent to apply our framework also for friend recommendations (i.e. Link Prediction), where the majority of earlier work infers new future interactions between users by mainly focusing on structural properties of a single type of network. Finally, 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).

8. REFERENCES

[1] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[2] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. on Information Systems*, 22(1):143–177, 2004.

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

[4] J. He, and W. Chu A Social Network-Based Recommender System (SNRS) In *Annals of Information Systems, Special Issue on Data Mining for Social Network Data*, Vol.12, pages 44–74, 2010.

[5] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. ACM SIGIR Conf.*, pages 230–237, 1999.

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

[7] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. on Information Systems*, 22(1):5–53, 2004.

[8] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proc. 4th ACM RecSys Conf.*, pages 135–142, 2010.

[9] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proc. ACM CIKM Conf.*, pages 247–254, 2001.

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

[11] H. Li, S. Bhowmick, and A. Sun. Affrank: Affinity-driven ranking of products in online social rating networks. *Journal of the American Society for Information Science and Technology*.

[12] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. 12th CIKM Conf.*, 2003.

[13] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In *Proc. Federated Int. Conf. on The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, pages 492–508, 2004.

[14] R. McLauglin and J. Herlocher. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proc. ACM SIGIR Conf.*, pages 329–336, 2004.

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

[16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering on netnews. In *Proc. Conf. Computer Supported Collaborative Work*, pages 175–186, 1994.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. WWW Conf.*, pages 285–295, 2001.

[18] P. Symeonidis, E. Tiakas, and Y. Manolopoulos. Transitive node similarity for link prediction in social networks with positive and negative links. In *Proc. 4th ACM RecSys Conf.*, pages 183–190, 2010.

[19] V. Vasuki, N. Natarajan, Z. Lu, and I. Dhillon. Affiliation recommendation using auxiliary networks. In *Proc. 4th ACM RecSys Conf.*, pages 103–110, 2010.

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