With a Little Help from My Friends (and Their Friends): Influence Neighborhoods for Social Recommendations

Avni Gulati
Computer Engineering Department
San Jose State University
San Jose, CA, USA
avni.gulati@sjsu.edu

Magdalini Eirinaki
Computer Engineering Department
San Jose State University
San Jose, CA, USA
magdalini.eirinaki@sjsu.edu

ABSTRACT

Social recommendations have been a very intriguing domain for researchers in the past decade. The main premise is that the social network of a user can be leveraged to enhance the rating-based recommendation process. This has been achieved in various ways, and under different assumptions about the network characteristics, structure, and availability of other information (such as trust, content, etc.) In this work, we create neighborhoods of influence leveraging only the social graph structure. These are in turn introduced in the recommendation process both as a pre-processing step and as a social regularization factor of the matrix factorization algorithm. Our experimental evaluation using real-life datasets demonstrates the effectiveness of the proposed technique.

CCS CONCEPTS

• Information systems → Social recommendation; **Social networks**; **• Human-centered computing → Collaborative filtering**; **Social recommendation**; **• Theory of computation** → Social networks.

KEYWORDS

social recommender systems, influence propagation, recommendation systems

ACM Reference Format:
Avni Gulati and Magdalini Eirinaki. 2019. With a Little Help from My Friends (and Their Friends): Influence Neighborhoods for Social Recommendations. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313745

1 INTRODUCTION

Humans form opinions and make multiple decisions, small or big, on any given day. For many of those, such as deciding which movie to watch, or which new restaurant to try, or even which graduate school to attend, we tend to rely not only on our own personal judgment, knowledge and intution, but also that of others, especially those whose opinion we value and trust. This type of influence is what social recommender systems attempt to simulate. Social recommender systems leverage social relations to improve the rating-based recommendation process [1], based on the assumption that a user's preferences are likely to be similar to, or influenced by these of their friends [11].

Most of existing related work is making the assumption that the users are mostly influenced by their direct neighbors. However, studies have shown that humans tend to form opinions not only through their direct connections, but also via acquaintances [8, 9]. This concept aligns with the study of influence propagation in social networks. This line of work is drawing inspiration from social correlation theories such as homophily and social influence [18, 19]. Most of the existing approaches regard this as a long-term, network diffusion process, and follow a graph-theoretic approach to solve the problem of identification of influentials in a social network [2, 6, 13].

In this work, we introduce a social recommendation algorithm that leverages the influence propagation beyond direct neighbors in a social graph. In particular, we explore the integration of the social graph as input to the recommendation process. We employ our threshold-bounded influence propagation algorithm [10] to generate social graph-based neighborhoods for each user. This neighborhood of influence is used as a pre-processing step. The outcome of this algorithm is also integrated as a social regularization factor in the matrix factorization process. Our experimental results using real-life datasets demonstrate the effectiveness of such an approach. The proposed high-level methodology is depicted in Figure 1.

The rest of the paper is organized as follows: we review the related work in Section 2. We then present the influence propagation algorithm in Section 3. The proposed recommendation algorithm with social regularization is presented in Section 4. Finally, in Section 5 we discuss the results of our experimental evaluation and conclude with our observations and our plans for future work in Section 6.

2 RELATED WORK

Social recommender systems have gained a lot of attention from the research in an effort to leverage social relationships to improve the recommendation process. Rooted in the sociology concepts of homophily and social influence [18], this line of work is based on the assumption that users' preferences are influenced more by these of their connected friends than these of unknown users [26]. Tang et al. [22] give a narrow definition of social recommendation as "any recommendation with online social relations as an additional

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313745

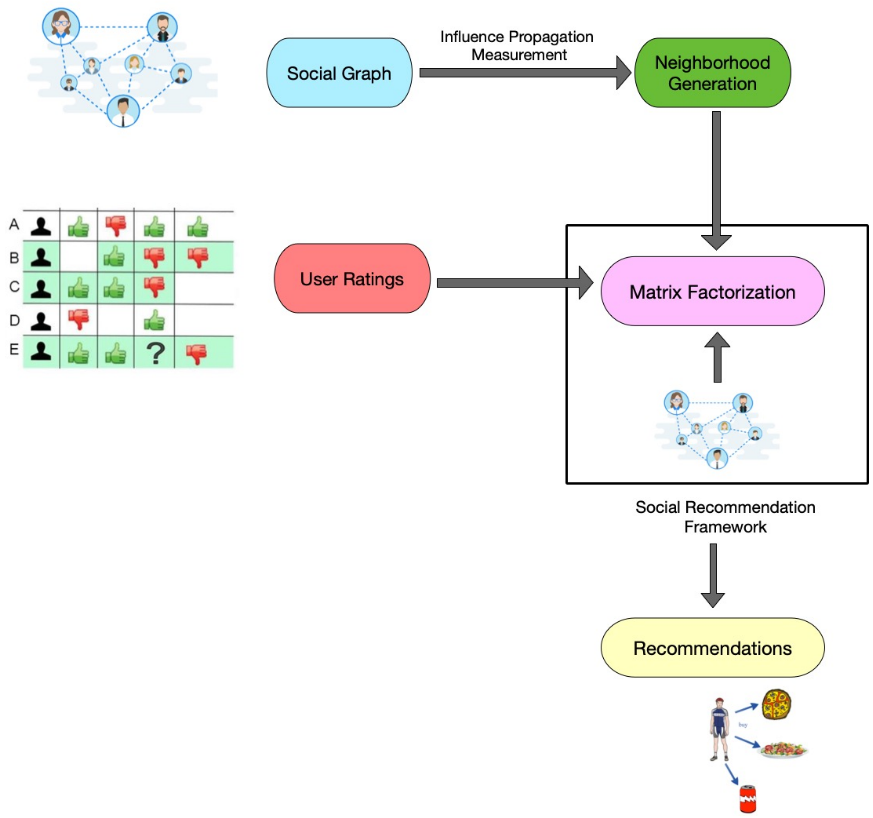

Figure 1: Social Recommendation Framework incorporating influence propagation approach for neighborhood generation in Matrix Factorization for generating effective recommendations for users.

input, i.e., augmenting an existing recommendation engine with additional social signals" (a broader definition, not applicable to this work, refers to recommender systems targeting social media domains [11]).

The various proposed approaches can be categorized depending on the type of social relationship (trust, friendship etc.), the type of the underlying recommendation algorithm (model-based, memory-based, etc.), and the level of integration of the social information in the recommendation process. A common approach is to enhance model-based recommender systems with social connections, again most often expressed as trust. This can be done through co-factorization, where the assumption is that the users share the same preference vector in both the rating and the social spaces (e.g. [20]), ensemble methods, where the resulting recommendation is derived by the linear combination of two systems (e.g. [15, 21]), or regularization, where priority is given to the social-based ratings (e.g. [12, 16]).

An alternative line of work involves ways to enhance the memory-based collaborative filtering process by forming the user's neighborhood using similarities deriving from the users' ratings and/or their social relationships, focusing on trust [5, 7, 17, 23–25]. One of the most recent works [4] introduces a Deep Learning Matrix Factorization (DLMF) approach to address the issues related to the initialization of latent feature vectors in Matrix Factorization (MF). The authors use the user's trust on other users who belong to his/her clique as the basis of neighborhood formation, and integrate it with Matrix Factorization as a trust regularization factor to predict ratings.

Ma et al. [16] introduced social regularization constraints in the recommendation framework. They used the social information to effectively predict the missing user-item matrix. Experimental results on real world data showed that their algorithm outperformed the traditional Matrix Factorization methodology. In this work we follow a similar approach, however we focus on influence as derived from the social graph connections, rather than metadata related to the users. We leverage our proposed influence propagation algorithm [10] and create social graph-based personalized neighborhoods that are subsequently used as input to the recommendation process. Moreover, we combine the social information of the users gained from influence propagation algorithm to Matrix Factorization forming a unified social recommendation framework.

3 NEIGHBORHOOD FORMATION USING INFLUENCE PROPAGATION

In this section we briefly present our Threshold-Bounded Influence Propagation algorithm (TB-IP) [10]. While this algorithm was introduced to solve the *min-max* problem of maximizing influence coverage on a social network with the minimum number of "seed" nodes, in this work we utilize it as a means to integrate social graph data to the recommendation process[1].

A social network can be modeled as a weighted-edge graph $G = (V, E, W)$, where V is the set of all the vertices in the graph (i.e. people), E is the set of edges (i.e. their connections), and W is the set of edge weights. An edge from node u to node v ($u \rightarrow v$) signifies that user u "follows" v in the social network, and the edge weight $w(u_v)$ represents the influence of v on u. Our objective is to identify the influential nodes $v \in M, M \subset V$ that influence the remaining nodes $u \in D, D \subset V$, $D \cap M = \emptyset$ such that $|M|$ is minimized and $|D|$ is maximized. The *Threshold-Bounded Influence Propagation in Digraph (TB-IP)* algorithm takes as input a graph G and outputs sets M and D.

The algorithm accepts the following as parameters: a) a threshold thr that is defined according to the selected threshold condition and is used to determine whether a node u is influenced by v, b) a maximum number of *"hops"* (i.e. the maximum allowed depth of influence propagation), c) a *decay factor* for the influence propagation, and d) a ranking strategy for initializing the nodes (e.g. PageRank, Out-degree centrality, Upper-bound PageRank etc.). We consider three alternatives as threshold conditions:

- *Condition I: No Threshold (NoThr):* The first condition is considered for two-hops in the graph considering no threshold for influence propagation.
- *Condition II: Average Threshold (AvgThr):* The second condition takes threshold as the average of edge-weights of the entire network. This would mean that the threshold is constant for all the nodes, depending on the characteristics of the network as a whole.
- *Condition III: Edge-Weight dependent Threshold (EWThr):* The third condition determines the threshold by taking average of edge-weights of all the outgoing-edges from the node in the graphically represented social network. Thus, this threshold condition is vertex-specific but constant for every node.

The algorithm begins by sorting all nodes in descending order based on their assigned rank $r(v)$. Then, beginning with the highest ranked node, it examines whether its direct connections should be added to its neighborhood and therefore be considered influenced or not. This is determined by examining whether the edge-weight $w(u_v)$ of a connected node u is above the set threshold thr or not. If at least one connected node qualifies, node v is being added to the "influencer" set M and the qualifying nodes are being added to the "influenced" set D. If the algorithm is set to examine nodes that are indirectly connected to v (depth is defined by the *maxhop* parameter), each of the nodes that were added to $N_G(v)$ in the previous step are used to find their directly connected nodes. However, in this case, the respective edge-weights are updated by the *hopping factor* before being evaluated against the threshold thr as follows:

$$w(p_u) * (1 + hop * decay) \geq threshold \tag{1}$$

where *hop* is an integer value 0 for immediate neighbors (i.e. adjacent nodes) and it increments by 1 for each subsequent hop, and *decay* is a constant equal to 0.1[2].

When a node satisfies the threshold condition and has not been previously added to the "influenced" set D, then it is being added to both this set, and the neighborhood of v, $N_G(v)$. The algorithm stops this loop when either the maximum depth (i.e. number of hops) has been reached, or no nodes are qualifying as "influenced" in the current level. This process is being repeated for each of the nodes, as selected from the ranked list, and as long as they have not already been added in the "influenced" set D. The above process is described in detail in Algorithm 1.

Algorithm 1 Threshold-Bounded Influence Propagation in Digraph (TB-IP)

Require: A weighted and directed social network $G = (V, E, W)$
Ensure: Influenced Vertices $D \subset V$ and Influential Vertices $M \subset V$a
Initialize: thr, $maxhop$, $visit = 0$, $hop = 0$, $decay = 0.1$, $D = \emptyset$, $M = \emptyset$
$\forall v \in V$, $r(v)$ = compute_rank(v)
Add all $(v_i, v_j) \in V$ to ordered set S s.t. if $r(i) > r(j)$ then $i < j$
for each $v \in S$ **do**
  if $w(u_v) > thr$ and $\exists\, u \in V \setminus D$ **then**
    $N_G(v) = N_G(v) \cup u$
    $D = D \cup u$
    $M = M \cup v$
    $visit = 1$
  end if
  while $hop \leq maxhop$ and $visit = 1$ **do**
    $hop = hop + 1$
    $visit = 0$
    for each $u \in N_G(v)$ **do**
      $w(p_u) = w(p_u) * (1 + hop * decay)$
      if $w(p_u) > thr$ and $\exists\, p \in V \setminus D$ **then**
        $N_G(v) = N_G(v) \cup p$
        $D = D \cup p$
        $visit = 1$
      end if
    end for
  end while
end for
output D, M

4 SOCIAL REGULARIZATION IN MATRIX FACTORIZATION

A lot of techniques have been used in the past to generate effective recommendations for users. Perhaps the most popular methodology in terms of both effectiveness and efficiency is matrix factorization [14]. This collaborative filtering technique takes as input the

[1]More details as well as the experimental evaluation of this algorithm can be found in [10].

[2]Value set after experimentation.

matrix containing user ratings for items and applies dimensionality reduction to identify latent features of correlated user-item interactions. Mathematically, the sparse user-item (or *utility*) matrix ($R \in \mathbb{R}^{m \times n}$) can be decomposed into two matrices, representing the users ($P \in \mathbb{R}^{k \times m}$) and items ($Q \in \mathbb{R}^{k \times n}$). The low-ranked matrix factorization approach approximates the utility matrix R by multiplication of k-rank factors (where $k < min(m, n)$):

$$\mathrm{R} \approx \mathrm{P}^{\mathsf{T}} \times \mathrm{Q} \tag{2}$$

Singular Value Decomposition (SVD) is traditionally used to factorize the observed ratings from the utility matrix to obtain the latent factors by minimizing the following objective function:

$$\min_{P,Q} \frac{1}{2} \sum_{u=1}^{m} \sum_{i=1}^{n} I_{ui} (R_{ui} - P_u^T \cdot Q_i)^2 + \lambda_1 (\|P\|^2) + \lambda_2 (\|Q\|^2), \tag{3}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are the regularization factors added to avoid overfitting. I_{ui} is an indicator function which is 1 when user u has rated item i and it is equal to 0 if the user has not rated the item. Several optimization approaches can be used to find local minimum of Equation 3, such as Alternating Least Squares (ALS) or Stochastic Gradient Descent (SGD).

In this work, we integrate the social graph data in the recommendation process following a similar approach to that of Ma et al. [16], who introduced a social regularization term in the objective function, incorporating the preferences of a user's friends, i.e. the directly connected nodes of a node in the social graph. We employ the TB-IP algorithm discussed in section 3 in two ways, as shown in Figure 1.

First, we apply the algorithm as a pre-processing step to define a social graph-based neighborhood of influence in the graph. This process removes some of the nodes that do not belong to the circle of influence of any other users, and thus are expected to contribute less to the rating prediction. Secondly and most importantly, we define a new social similarity function that integrates the edge-weights derived from the TB-IP algorithm as social relation weights between users.

The problem of estimating the utility matrix R is therefore reduced to minimizing the following objective function and regularization terms:

$$\begin{aligned} \min_{P,Q} \mathcal{L}_2(R, P, Q) = \frac{1}{2} \sum_{u=1}^{m} \sum_{i=1}^{n} I_{ui} (R_{ui} - P_u^T \cdot Q_i) + \\ \frac{\beta}{2} \sum_{v \in N_G^{+(\mathrm{u})}} SocSim(u, v) (\|P_u - P_v\|^2) + \\ \lambda_1 (\|P\|^2) + \lambda_2 (\|Q\|^2), \end{aligned} \tag{4}$$

where $\beta > 0$, and $N_G{}^{+(u)}$ denotes u's out-link connections belonging to its neighborhood N_G^u as derived by the TB-IP algorithm.

We further augment this social regularization term by introducing a social similarity to weigh how much each neighbor's preference affects the user's predictions. We define the social similarity $SocSim(u, v)$ between two users u and v as follows:

$$SocSim\,(u, v) = \frac{1}{2} (Sim\,(u, v) + w(u_v)). \tag{5}$$

$Sim(u, v)$ stands for any vector similarity metric that can be employed to calculate how similar the two user vectors u and v are (in terms of similar ratings). In this work, we use the Pearson correlation coefficient [3] formulated as:

$$Pearson\,(u, v) = \frac{\sum\limits_{i \in P_u \bigcap P_v} (R_{ui} - \bar{R}_u)(R_{vi} - \bar{R}_v)}{\sqrt{\sum\limits_{i \in P_u \bigcap P_v} (R_{ui} - \bar{R}_u)^2} \sqrt{\sum\limits_{i \in P_u \bigcap P_v} (R_{vi} - \bar{R}_v)^2}}. \tag{6}$$

Here, i is the subset of total items user u and user v have both rated. R_{ui} is the rating given by user u to item i. $\bar{R}_u$ and $\bar{R}_v$ are the average ratings of users u or v respectively and R_{ui}, R_{vi} are the ratings of users u and v respectively for item i.

The second term, $w(u_v)$, is the edge weight derived by Algorithm 1. This weight measures the extend to which user v influences user u, and is based on the social graph's structure. Therefore, this social similarity metric leverages both the social graph and the rating preferences of the users.

A local minimum of Equation 4 can be obtained by applying SGD on the P_u and Q_i feature vectors, solving the partial derivatives on both the feature vectors as shown in Equations 7 and 8:

$$\begin{aligned} \frac{\partial \mathcal{L}_2}{\partial P_u} = \sum_{i=1}^{n} I_{ui} (P_u^T \cdot Q_i - R_{ui}) Q_i + \lambda_1 P_u \\ + \beta \sum_{v \in N_G^{+(\mathrm{u})}} SocSim(u, v)(P_u - P_v) \end{aligned} \tag{7}$$

$$\frac{\partial \mathcal{L}_2}{\partial Q_i} = \sum_{u=1}^{m} I_{ui} (P_u^T \cdot Q_i - R_{ui}) P_u + \lambda_2 Q_i \tag{8}$$

5 EXPERIMENTAL EVALUATION

The need for datasets that combine both user-item-rating triplets and social graph data for the users limits our choices to a big extend. One of the most known and broadly used datasets that combines both sources of information is the Yelp Challenge dataset[3]. We also employ the Epinions dataset[4]. Epinions is a trust-based rating dataset - in essence users have denoted whether they trust other users in their reviews and ratings or not. For our work we have translated "trust" endorsements to directed edges. While the resulting graph is not representing social relationships (of friendship), it encapsulates some social characteristics (users trusting other users in some context), and in the absence of other publicly available datasets, we follow the example of related research approaches and use this as a second real-life data source.

For the Yelp dataset we select the Las Vegas subset, including only the users who had rated at least one business in the city of Las Vegas. This is because the Yelp social subset (i.e. friends's network) is very sparse, with the Las Vegas dataset containing the most dense social graph subset. Hence, we only considered those connections who rate restaurants and reside in the same area. The details of the social graph of each dataset are included in Table 1, while the user-item-rating characteristics are included in the first row of Tables 2 and 3.

[3]https://www.yelp.com/dataset/challenge
[4]www.epinions.com

For each dataset, we generate recommendations using two baselines and several variations of our proposed methodology. We employ the optimal parameters for the TB-IP algorithm, as experimentally identified in [10], running the algorithm using the *EWThr* threshold selection strategy for two *hops* (named *TwoD EWThr*).

Table 1: Social graph dataset characteristics

Dataset	Yelp (Las Vegas)	Epinions (Trust)
# Nodes	247,111	18,089
# Edges	5,340,568	355,217
Avg. Degree	21.612	19.6372
Type	Directed	Directed

Table 2: Yelp input datasets

Dataset	# Users	# Businesses	# Ratings
BaseAll	247,111	26,304	1,104,768
TwoD EWThr	96,900	24,632	733,408

Table 3: Epinions input datasets

Dataset	# Users	# Items	# Ratings
BaseAll	22,164	296,277	922,267
TwoD EWThr	7,615	222,281	539,963

We evaluate the effect of the neighborhood formation in two levels: i) as the input (i.e. as a pre-processing step) to the recommendation process, and ii) as an integral part of the recommendation algorithm.

To evaluate the first objective, we apply TB-IP to the entire dataset. We therefore create two different input sets, as shown in Tables 2 and 3:

Dataset 1: BaseAll. This dataset has the entire user-item-rating data.

Dataset 2: TwoD EWThr. This dataset has the rating data only for users obtained after applying the TB-IP Algorithm with *EWThr* threshold condition. We use the *Outdeg* ranking methodology for initial ranking of nodes.

We consider two baseline methodologies to evaluate our proposed Social Recommendation Framework. The first baseline evaluates the traditional Matrix Factorization (*SimpleMF*) explained in equation 3. As the second baseline we consider the Social Regularization in Matrix Factorization (*Simple SoReg*) developed by Ma et al. [16]. In that work, the similarity function does not incorporate the social graph weight (in essence reflects a normalized Pearson correlation coefficient). We compare these baselines against several variations of our method, employing the TB-IP algorithm and the new similarity metric function (*SocSim*) that we defined in Equation 5. The different setups are explained in what follows.

Experiment 1: SimpleMF (First Baseline). Using as input the entire dataset (*BaseAll*) and implemented *Basic MF*, i.e. traditional Matrix Factorization [14].

Experiment 2: TB-IP SimpleMF. In this experiment, we considered the social-graph induced dataset (*TwoD EWThr*) and generated recommendations using *Basic MF*.

Experiment 3: Simple SoReg (Second Baseline). Here, we evaluated *SoReg* methodology with *Sim* for *BaseAll* dataset, as discussed in [16].

Experiment 4: TB-IP SoReg. We use as input the social-graph induced dataset *TwoD EWThr* and generate recommendations using *SoReg* technique using the normalized Pearson similarity metric.

Experiment 5: SocSim Simple SoReg. Here, we apply our proposed *SoReg* algorithm incorporating the *SocSim* similarity metric over the entire (*BaseAll*) dataset.

Experiment 6: SocSim TB-IP SoReg. In this experiment, the preprocessing step is applied to create the input dataset *TwoD EWThr* and the recommendations are generated by using the *SoReg* technique with *SocSim* similarity metric.

We performed 5-fold cross validation for all the experimental setups.[5] We evaluated all combinations of the following Matrix Factorization parameters: *rank (f)* = $\{10, 20\}$, *regularization* (λ) = $\{0.1, 0.01, 0.001\}$, *maximum iterations (max_iter)* = $\{10, 20\}$. For the Social Regularization technique we also used another metric, called *social regularization* (β) = $\{0.1, 0.01\}$.

We use RMSE and MAE evaluation metrics. The results, along with the optimal parameters for each setup, are presented with the corresponding experiment in Tables 4 and 6 (RMSE) and Tables 5 and 7 (MAE) for the Epinions and Yelp dataset respectively.

Table 4: RMSE and optimal parameter settings for different experiments performed for Epinions

Experiment	RMSE	f	λ	β	max iter
SimpleMF	1.063	20	0.001	0	10
TB-IP SimpleMF	1.046	20	0.001	0	10
Simple SoReg	1.046	20	0.01	0.1	20
TB-IP SoReg	**1.030**	20	0.01	0.1	20
SocSim Simple SoReg	1.061	20	0.01	0.1	10
SocSim TB-IP SoReg	1.043	20	0.01	0.1	10

Table 5: MAE and optimal parameter settings for different experiments performed for Epinions

Experiment	MAE	f	λ	β	max iter
SimpleMF	0.812	20	0.001	0	10
TB-IP SimpleMF	0.800	20	0.001	0	10
Simple SoReg	0.807	20	0.01	0.01	10
TB-IP SoReg	**0.794**	20	0.01	0.01	20
SocSim Simple SoReg	0.813	20	0.01	0.1	10
SocSim TB-IP SoReg	0.799	20	0.01	0.01	10

We observe that the *TB-IP* prefixed approaches, i.e. those that include the pre-processing step of identifying the

[5]We used ReQ python library from https://github.com/Coder-Yu/RecQ to perform our experiments on the baselines and updated the code to reflect our methodology.

Table 6: RMSE and optimal parameter settings for different experiments performed for Yelp

Experiment	RMSE	f	λ	β	max iter
SimpleMF	1.252	20	0.001	0	10
TB-IP SimpleMF	1.175	20	0.01	0	10
Simple SoReg	1.222	20	0.01	0.01	10
TB-IP SoReg	1.152	20	0.01	0.01	10
SocSim Simple SoReg	1.212	20	0.01	0.1	10
SocSim TB-IP SoReg	**1.145**	20	0.01	0.1	10

Table 7: MAE and optimal parameter settings for different experiments performed for Yelp

Experiment	MAE	f	λ	β	max iter
SimpleMF	0.970	20	0.001	0	10
TB-IP SimpleMF	0.909	20	0.01	0	10
Simple SoReg	0.951	20	0.01	0.01	10
TB-IP SoReg	**0.895**	20	0.01	0.01	10
SocSim Simple SoReg	0.953	20	0.01	0.1	10
SocSim TB-IP SoReg	0.899	20	0.01	0.01	10

influential neighborhoods within the original dataset, outperform the others in both datasets. For the Epinions dataset we observe that the *TB-IP SoReg* approach outperforms the others, giving the best results both in terms of RMSE and MAE, with second best our *SocSim TB-IP SoReg* approach. On the other hand, we observe that for the much larger Yelp dataset, our approach (*SocSim TB-IP SoReg*) gives better results than the other approaches in terms of RMSE while having very comparable MAE results with the *TB-IP SoReg*. In fact, we observe a big improvement over the baselines, ranging from 6.3% to 8.6% for RMSE and 5.8% to 7.7% for MAE.

As previously mentioned, the Yelp social graph represents a real social network (in that the edges reflect friendships between users), compared to the one of Epinions that signifies trust relationships ("I trust someone's opinions/reviews") but not necessarily real-life friendships or social relationships. Therefore we can conclude that the difference in our algorithm's effectiveness lies in the fact that it works better when real-life relationships are represented in the social graph and therefore better reflected in the social weights $w(u_v)$ and the derived *SocSim* similarity.

Tables 8 and 9 show the running time for neighborhood formation (where applicable) and recommendation for the Epinions dataset. The times reported reflect the optimal parameters for each setup. As expected, the proposed approaches introduce an overhead for the neighborhood formation phase, resulting in higher total times. However, this pre-processing step is performed offline and is not expected to affect real-time recommendations. On the contrary, we observe that both approaches integrating our social regularization factor with the *SocSim* similarity are faster than their simple Pearson similarity counterparts during recommendation time. This is due to the fact that the algorithm took less iterations to converge in the former case. We hypothesize that this is due to the fact that the social graph weights $w(u_v)$ introduced in the user similarity (*SocSim*) act as a form of better initialization for the matrix factorization process, helping the algorithm to converge in fewer steps.

Overall our results confirm our claim that combining neighborhood formation with influence propagation and social regularization produces better recommendations than the traditional social recommendation approaches.

Table 8: Epinions time analysis for least RMSE (in min)

Experiment	Neigh	Rec	Total
SimpleMF	0	9	9
TB-IP SimpleMF	26	4	30
Simple SoReg	0	53	53
TB-IP SoReg	26	19	45
SocSim Simple SoReg	0	10	10
SocSim TB-IP SoReg	26	13	39

Table 9: Epinions time analysis for least MAE (in min)

Experiment	Neigh	Rec	Total
SimpleMF	0	9	9
TB-IP SimpleMF	26	4	30
Simple SoReg	0	28	28
TB-IP SoReg	26	19	45
SocSim Simple SoReg	0	10	10
SocSim TB-IP SoReg	26	7	33

6 CONCLUSIONS

In this work, the problem of generating social recommendations is explored in the context of influence propagation. We employ an influence propagation algorithm to create social graph-based neighborhoods of influence. This neighborhood is then used as input to the recommendation algorithm, both as a pre-processing step and as an integral part of the recommendation algorithm in the form of a similarity metric. Our experiments show that the integration of social graph-derived information improves the traditional rating-based recommendation process. Moreover, we observe that in the case of real-life relationships reflected in the input social graph, our proposed similarity metric greatly improves the results and accelerates the recommendation algorithm.

Part of our ongoing work includes improving further on the social regularization component and social similarity metric. We are also exploring employing deep learning to initialize the latent vectors, and want to study how other influence propagation algorithms and centrality metrics can be employed as a means to create user influence neighborhoods. Finally, we believe that the introduction of a social component in the recommendation process not only improves the accuracy of recommendations, but can also be leveraged to generate explainable recommendations. This is an interesting extension of this work that we plan to explore in the near future.

REFERENCES

[1] C.C. Aggarwal. 2016. *Recommender Systems: The Textbook* (1st ed.). Springer Publishing Company, Incorporated.

[2] Christian Borgs, Michael Brautbar, Jennifer T. Chayes, and Brendan Lucier. 2014. Maximizing Social Influence in Nearly Optimal Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*. 946–957.

[3] John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 43–52.

[4] ShuiGuang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. 2017. On Deep Learning for Trust-Aware Recommendations in Social Networks. *IEEE Trans. Neural Netw. Learning Syst.* 28, 5 (2017), 1164–1177.

[5] Magdalini Eirinaki, Malamati D. Louta, and Iraklis Varlamis. 2014. A Trust-Aware System for Personalized User Recommendations in Social Networks. *IEEE Trans. Systems, Man, and Cybernetics: Systems* 44, 4 (2014), 409–421.

[6] Magdalini Eirinaki, Nuno Moniz, and Katerina Potika. 2016. Threshold-Bounded Influence Dominating Sets for Recommendations in Social Networks. In *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom), BDCloud-SocialCom-SustainCom 2016, Atlanta, GA, USA, October 8-10, 2016*. 408–415.

[7] Jennifer Golbeck. 2006. Trust on the world wide web: a survey. *Found. Trends Web Sci.* 1, 2 (2006), 131–197.

[8] Mark Granovetter. 2018. *Getting a job: A study of contacts and careers*. University of Chicago press.

[9] Mark S Granovetter. 1977. The strength of weak ties. In *Social networks*. Elsevier, 347–367.

[10] Avni Gulati and Magdalini Eirinaki. 2018. Influence Propagation for Social Graph-based Recommendations. In *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*. 2180–2189.

[11] Ido Guy. 2015. *Social Recommender Systems*. 511–543.

[12] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*. 135–142.

[13] David Kempe, Jon Kleinberg, and Éva Tardos. 2015. Maximizing the Spread of Influence through a Social Network. *Theory of Computing* 11, 4 (2015), 105–147.

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37.

[15] Hao Ma, Irwin King, and Michael R. Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*. 203–210.

[16] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*. 287–296.

[17] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys 2007, Minneapolis, MN, USA, October 19-20, 2007*. 17–24.

[18] James M. Cook Miller McPherson, Lynn Smith-Lovin. 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology* 27 (2001), 415–444.

[19] Cosma Rohilla Shalizi and Andrew C. Thomas. 2011. Homophily and Contagion Are Generically Confounded in Observational Social Network Studies. *Sociological Methods Research* 40, 2 (5 2011), 211–239.

[20] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. 2013. Exploiting homophily effect for trust prediction. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*. 53–62.

[21] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: discerning multi-faceted trust in a connected world. In *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*. 93–102.

[22] Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. *Social Netw. Analys. Mining* 3, 4 (2013), 1113–1133.

[23] Iraklis Varlamis, Magdalini Eirinaki, and Malamati D. Louta. 2010. A Study on Social Network Metrics and Their Application in Trust Networks. In *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2010, Odense, Denmark, August 9-11, 2010*. 168–175.

[24] Iraklis Varlamis, Magdalini Eirinaki, and Malamati D. Louta. 2013. Application of Social Network Metrics to a Trust-Aware Collaborative Model for Generating Personalized User Recommendations. In *The Influence of Technology on Social Network Analysis and Mining*. Springer Vienna, 49–74.

[25] Patricia Victor, Martine De Cock, and Chris Cornelis. 2010. Trust and Recommendations. In *Handbook of Recommender Systems*. springer, 645–675.

[26] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. TwitterRank: finding topic-sensitive influential twitterers. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*. 261–270.