San Jose State University

From the Selected Works of Magdalini Eirinaki

January 24, 2019

Influence Propagation for Social Graph-based Recommendations

Avni Gulati, *San Jose State University*

Magdalini Eirinaki, *San Jose State University*

Available at: https://works.bepress.com/magdalini_eirinaki/65/

Influence Propagation for Social Graph-based 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 networking is an inevitable behavior of humans living in a society. In recent years, and with the rise of online social networks, personalized recommendations that leverage the social aspect have become a very intriguing domain for researchers. In this work, we explore how influence propagation and the decay in the cascading effect of influence from influential users can be leveraged to generate social graph-based recommendations. Understanding how influence propagates within a social network is itself a challenging problem. Few researchers have considered influence propagation and even fewer have considered decay in the cascading effect of influence in a social network. In this work we model the decay in influence propagation in directed graphs, utilizing the structural properties of the social graph to measure the propagated influence beyond one-hop. We then employ this influence propagation model to form social recommendations, and present our experimental results using real-life datasets.

Keywords—online social networks; influence propagation; social recommender systems; influence propagation

I. INTRODUCTION

Information is a very powerful tool. The information gathered from our friends help us form opinions and we take decisions based on those opinions. Not only are we influenced by our friends, but friends of friends and even acquaintances influence our decision making. This idea was first introduced by Mark Granovetter in 1960s where he conducted a study by interviewing people who had recently changed jobs [1]. He found that most of these people had heard about a job opening through an acquaintance and very few got the jobs with the help of personal contacts. He conceptualized this influence from acquaintances as "strength of weak ties" [2], where strong ties are among immediate friends while weak ties are when two people do not directly know each other, but have friends in common. Granovetter's sociological concepts can be applied to the current online social networks as well. Identifying only a few people who can influence the large majority of the remaining people in the network can be a powerful tool in maximizing social outreach in a social network. This information can be effectively utilized, for example, in the advertising industry by targeting a few popular celebrities on social media to advertise items among the masses, or by government agencies in order to reach the maximum number of people in disease outbreaks and emergencies.

Social recommender systems leverage social relations to improve the rating-based recommendation process [3], based on the assumption that a user's preferences are likely to be similar to, or influenced by these of her friends [4]. Most of existing related work is making the assumption that the users are mostly influenced by their direct neighbors. In this work, we explore the concepts of influential users and influence propagation beyond direct neighbors in the context of social recommender systems.

The focus of this work is two-fold. As a first task, we utilize the structural characteristics of a directed graph to find primary *influentials* and the nodes they *influence*. The objective is to *maximize* influence with the *minimum* number of influencers, and thus this is defined as a *min-max* problem. We extend our previous work that focused on undirected graphs with no cascades [5], and propose a threshold-bounded influence propagation algorithm that can be applied on directed graphs taking into account cascading of information in the network. We then employ our proposed algorithm as a pre-processing step in the context of a social recommender system by forming social graph-based user neighborhoods to be used as input to the recommendation process. This methodology is depicted in Figure 1.

The rest of the paper is organized as follows: we review the related work in the two areas this work touches upon, namely identification of influential users and social recommender systems, in Section II. After defining our objectives in Section III, we discuss in detail our proposed methodology and algorithm for identifying influential users in Sections II and V. Finally, in Section VI we discuss the results of our two-step experimental evaluation of the proposed algorithm, first in the context of influence propagation and social network coverage, and subsequently by indirectly evaluating it as a preprocessing step in the recommendation process and conclude with our plans for future work in Section VII.

II. RELATED WORK

A. Identification of influential users

The problem of identifying influentials has gained a lot of attention from several research communities as it has applications in viral marketing [6], [7], [8], disease prevention [9] and propagation [10], politics [11], [12], smart cities

Fig. 1. Neighborhood formation by measuring influence propagation in social network to generate recommendations for users

[13], software sharing frameworks [14], etc. There are several ways to approach this problem and a few assumptions to be made when one regards this as a graph theoretic one, including whether the graph is directed or not, initialization of the graph, levels of influence propagation and any decay factors associated with it, as well as conditions to activate (i.e. “influence”) nodes. This has resulted in a diverse body of research works addressing this problem.

Identifying the most influential users of a market was first studied as an algorithmic problem by Domingos and Richardson [15]. They apply data mining techniques to viral marketing, by modeling markets as social networks. They study the spread of influence using probabilistic models of interactions. Each vertex is associated with a value that quantifies how much she can influence other vertices and is used to optimally determine which vertices to choose as influentials. In their empirical study, using the EachMovie database, their proposed market strategy performs much better than two simple existing strategies.

A more recent work is identifying influentials by calculating pairwise Spearman correlation between different influence metrics. The authors draw an analogy of Github with Twitter and perform analysis of correlation between follow, mention and retweets in Twitter [16]. Other approaches include using the simulated annealing (SA) algorithm to find top-k influential nodes in a social network. Jiang et al. [17] utilized SA to address the influence overlapping problem. A constraint simulated annealing methodology has been exploited to improve the performance of finding top-k influentials by considering influence loss when certain top nodes fail to function as expected [18]. Two-hop and three-hop problems in social network are explored in Gong et al. [19] using particle swarm optimization, a concept which was first introduced by Eberhart and Kennedy [20].

In this work we leverage the insights and results of our previous work [5], where we introduced the *NewGreedy*

algorithm to identify influentials in an undirected graph. The premise was that a node is influenced if a number of direct neighbors are “activated” (i.e. become influenced/are influentials) and therefore we assumed that no influence propagation took place. While we follow a similar approach, in this work we focus on directed graphs and assume that influence propagates further than one-hop neighbors. These assumptions pose different challenges and require a novel methodology to be addressed.

Kempe et al. [21] identified the optimization problem of selecting the most influential nodes in a social network as NP-hard. They proposed submodular approaches in social network diffusion model: *Linear Threshold Model* and *Independent Cascade Model*. In Linear Threshold Model, a node v randomly chooses threshold between 0 and 1 and gets activated when the combined effect from its neighbors exceeds the threshold value, only considering neighbors that were active in the previous iteration. This way they dynamically change the threshold value with each iteration. In Cascade Model, a node u can activate node v with a probability taking *neighbors* that have already tried and failed to activate v . In an Independent Cascade Model this probability is independent of neighbors. Motivated by the success of the above models, we also model thresholds dynamically, determined individually for each vertex, as a condition for node activation.

The importance of influence propagation for undirected graphs is well explained by Hangal et al. [22]. They have experimentally shown that the most influential path is more effective compared to the shortest path using DBLP and Twitter datasets by incorporating directed and weighted influence edges in a social graph. They defined a person as influential if he or she has high influence on many people. They also conceptualized the influence of a node as the sum of the influences the node has on others. The most influential path between two nodes was calculated by natural adaptation of Dijkstra’s algorithm. In this work we introduce a decay factor

for influence propagation so that it decreases after every hop in the path in a directed network.

B. Social recommender systems

Social recommender systems have gained a lot of attention from the research in an effort to leverage social relationships to improve the recommendation process. 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 [8], rooted in the sociology concepts of homophily and social influence [23]. Tang et al. [24] give a narrow definition of social recommendation as "any recommendation with online social relations as an additional 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 [25]).

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. [26]), ensemble methods, where the resulting recommendation is derived by the linear combination of two systems (e.g. [27], [28]), or regularization, where priority is given to the social-based ratings (e.g. [29], [30]).

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 [31], [32], [33], [34], [35], [36], [37].

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 and create social graph-based personalized neighborhoods that are subsequently used as input to the recommendation process.

III. PROBLEM DEFINITION

We define our problem as a *min - max* one: identify the minimum number of most influential nodes (called "seed" nodes), that can influence the maximum number of the remaining nodes (if not the entire network). We then proceed by claiming that this information is leveraged to generate social graph-based recommendations that are more accurate than traditional rating-based ones. Our research objectives can be further divided into the following sub-categories:

- *Defining edge-weights for the directed network:* We employ structural characteristics of the two nodes forming the edge to determine edge weights. The edge-weights defined are then employed to quantify the influence

propagated along vertices in the graphically represented social network.

- *Determining condition for influence propagation:* We utilize vertex-dependent threshold values in order to determine the condition of whether a node has been influenced by another node or not.
- *Ranking nodes:* Ranking of nodes is a very critical aspect for determining the top seed users. In this work, several existing methodologies for determining top-k nodes are employed to determine the ones that give optimal results during experimental analysis.
- *Neighborhood Generation:* We employ the influence propagation approach to generate neighborhood for generating recommendations for users. This results in personalized social network-based subset of users, and in turn, in more accurate recommendations.

These objectives are discussed in detail in the following sections.

IV. METHODOLOGY FOR MEASURING INFLUENCE PROPAGATION

We define our problem as follows: Identify the minimum number of seed *influentials*, so that there can be maximum number of *influenced* nodes in the social network. In this section we describe in detail our approach to this problem.

A. Determining Edge Weight in Directed Graphs

A social network can be modeled as a weighted-directed 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. In the directed graph used here, an edge from node u to node v ($u \rightarrow v$) signifies that user u "follows" v in the social network.

Edge weights are determined in the network by structural characteristics of the two nodes forming the edge. For instance, for the neighboring nodes u and v , the edge weight $w(u_v)$ represents the influence of v on u for this connection, and is defined as:

$$influence(u_v) = importance(v) / outdegree(u) \quad (1)$$

where $importance(v)$ is the *indegree centrality* of v . Here, v is *influential* and u is considered an *influenced* node only if $w(u_v) \geq threshold$ value where $threshold$ is a parameter of the algorithm (several strategies for determining its value are discussed in section IV-B). If this condition holds true, then for generating recommendations v is the seed node and u is considered in v 's neighborhood, i.e. $u \in N_G(v)$.

To demonstrate the cascading effect of influence being spread from a single node v , another node p is added that follows u where $w(p_u)$ is the respective edge-weight. This can be graphically represented as $p \rightarrow u \rightarrow v$. Node p is at two degrees of separation from v . Hence, if u is influenced by v , then we claim that node p is also influenced by v (and therefore $p \in N_G(v)$) if

$$w(p_u) * hopping\ factor \geq threshold \quad (2)$$

Here, we have introduced a *hopping factor* for capturing the essence of decay in information or influence with increasing hops which has been defined as:

$$\text{hopping factor} = 1 + \text{hop} * \text{decay} \quad (3)$$

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 0.1¹.

B. Determining condition for Influence Propagation

In the previous section, *threshold* was introduced to determine if *u* is influenced by *v*. Here, three threshold conditions are proposed for finding if a node is influenced:

- *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.

C. Methodologies for ranking nodes

This research intends to find the optimal ranking strategy for nodes that initiate the process of influence propagation in social graph *G* (mentioned in section IV-A). There are several ways proposed in the literature to identify important nodes in a graph. Most of those approaches use a graph-based metric such as centrality (degree, eigenvector, etc.) or PageRank to generate an initial ranking of nodes. Here, three methodologies for ranking of nodes are employed:

- *PageRank (PR)*: Page et al. [38] introduced PageRank for ranking importance of web pages. This is a very popular methodology used in social networks for ranking nodes.
- *Out-degree Centrality (Outdeg)*: Out-degree centrality measure is utilized here for initializing nodes to measure influence outreach in the social graph.
- *Upper-Bound PageRank (UB-PR)*: Liu et al. [39] used PageRank to find authority of nodes. They evaluated upper bounds of PageRank to find top authorities and introduced an efficient way to rank nodes. Here, this upper bound ranking is employed as one of the initial ranking criteria for nodes in the social network.

The process of identifying influential nodes is performed in sequential order starting with the top ranked nodes. These nodes are obtained from one of the ranking methodologies stated above. For each influential node considered in this list, influenced nodes are identified by following the methodology

mentioned in section VI-A. All the nodes that have been influenced once are removed from consideration of getting influenced by the successive nodes.

V. ALGORITHM

We follow the notation introduced previously and model the social network as a weighted-directed graph $G = (V, E, W)$, 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. In this section we introduce the *Threshold – Bounded Influence Propagation in Digraph* (TB-IP) that takes as input a graph *G* and outputs sets *M* and *D*. The algorithm accepts as parameters the following: a) a threshold *thr* that is defined according to the selected threshold condition (as described in Section IV-B), 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 (as described in Section IV-C).

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*, as defined in Section IV, before being evaluated against the threshold *thr*. 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.

As we can see in Algorithm 1, we perform a pass over all nodes in the social graph as any node can be a potential *influential*. Once a node has been *influenced*, we do not revisit it. This continues until the maximum number of hops is reached. Hence the running time of TB-IP algorithm is $O(nm)$ for $n = |V|$ and $m = |E|$. For experimental purposes, we limit the *maxhop* to 2 or 3 in this research work.

¹ Value set after experimentation

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

Input: A weighted and directed social network $G = (V, E, W)$

Output: Influenced Vertices $D \subset V$ and Influential Vertices $M \subset V$

```
1: Initialize:  $thr, maxhop, visit = 0, hop = 0, decay = 0.1, D = \emptyset, M = \emptyset$ 
2:  $\forall v \in V, r(v) = \text{compute\_rank}(v)$ 
3: Sort vertices  $v \in V$  in non-increasing order based on their rank  $r(v)$ 
4: for each  $v \in V$  in that order do
5:   if  $w(u_v) > thr$  and  $\exists u \in V \setminus D$  then
6:      $N_G(v) = N_G(v) \cup u$ 
7:      $D = D \cup u$ 
8:      $M = M \cup v$ 
9:      $visit = 1$ 
10:  end if
11:  while  $hop \leq maxhop$  and  $visit = 1$  do
12:     $hop = hop + 1$ 
13:     $visit = 0$ 
14:    for each  $p \in N_G(u)$  do
15:       $w(p_u) = w(p_u) * (1 + hop * decay)$ 
16:      if  $w(p_u) > thr$  and  $\exists p \in V \setminus D$  then
17:         $N_G(v) = N_G(v) \cup p$ 
18:         $D = D \cup p$ 
19:         $visit = 1$ 
20:      end if
21:    end for
22:  end while
23: end for
24: output  $D, M$ 
```

VI. EXPERIMENTAL EVALUATION

Our evaluation focuses on two main objectives: a) evaluate the proposed TB-IP Algorithm, and b) evaluate the social recommender system that leverages it.

A. TB-IP Algorithm

In order to evaluate our algorithm we employed three real-world social network datasets that are being broadly used in similar experimental setups, namely Epinions [40], AstroPhysics [41] and Yelp². Epinions is an online social network that has product reviews by the customers. It is a directed trust network where customers express their trust or distrust to other reviewers in the network. AstroPhysics is a scientific collaborative network among the authors that have published their papers in AstroPhysics domain. It is an undirected network which is changed into a directed graph for experimentation purposes by adding one edge for each direction. Yelp is a business review and recommendation service where users primarily review restaurants and rate them. This dataset is taken from the 2018 Yelp Challenge. It is

a directed network where users can be "friends" with each other. Since the Yelp social subset (i.e. the friends' network) is very sparse, we only considered users who had rated at least one business in the city of Las Vegas, one of the most dense social graph subsets. For these users, we considered only those connections who rate restaurants and reside in the same area. We will refer to this as the Yelp Las Vegas dataset. The network properties of the three datasets are given in Table I.

TABLE I
DATASET CHARACTERISTICS

Dataset	Astro Physics	Epinions	Yelp (Las Vegas)
# Nodes	18772	75879	247111
# Edges	198050	508837	5340568
Avg. Degree	21.10	6.706	21.612
Type	Undirected	Directed	Directed

As mentioned earlier, this research extends the *NewGreedy* algorithm [5]. The *EWThr* threshold approach is equivalent to finding activated nodes evaluated by the *NewGreedy* approach, where the threshold in an undirected graph is degree-dependent for every node. In that work, the threshold changed dynamically by removing the nodes in the graph that were already activated. In the methodology proposed in this paper, the *EWThr* condition for each node in a directed graph does not change dynamically but remains constant (yet can be vertex-specific). However, since the edge-weights determined are degree-dependent here, an analogy can be formed of this methodology with the *NewGreedy* algorithm.

In order to compare and draw conclusions easily, we replicated the setup used in the experimental evaluation of [5]. Therefore, for each of the three datasets, we generated subgraphs by randomly picking nodes as seed nodes and adding their neighbors so that the graph has more than 100 vertices but less than 600 vertices. These subgraphs were used as input to our experiments. Moreover, we set random seed equal to 1 for generating consistent results every time the algorithm runs.

Our objective is to compare the various ranking methodologies and threshold strategies in terms of the defined *min-max* problem. In other words, a setup is preferable when we are able to influence the maximum percentage of nodes with the minimum number of influencers. The results for the three datasets, namely AstroPhysics, Epinions, and Yelp, are shown in Figures 2, 3 and 4 respectively. In each figure, the top row shows the percentage of influenced nodes for different sizes of graphs (so higher is better), whereas the bottom row shows the respective number of influencers needed for the respective coverage (therefore lower is better). The first column, corresponding to the *No Threshold* condition (*NoThr*) is regarded as our naive baseline. This approach assumes that influence propagates for all neighboring nodes of any node in the graph, and up to two degrees of separation (regardless of this node's initial influence score). This approach is expected to achieve maximum coverage in very few steps, especially for small and highly connected graphs. However, this is a naive approach as it assumes that all nodes spread influence equally,

²<https://www.yelp.com/dataset/challenge>

an assumption that does not usually hold in social networks. However, we include here the results and use it as a baseline in order to assess the remaining two conditions, namely *Average Threshold (AvgThr)* and *Edge-Weight dependent Threshold (EWThr)*.

Indeed, we observe that in all networks the *NoThr* condition achieves very high (or maximum) network coverage with just a few influencer nodes. As implied previously, this was expected by design. As the algorithm starts with the highest ranked nodes and ranking is employing (outdegree or PageRank) centrality, the most central nodes cover a huge percentage of the graph in two hops, especially for such small subgraphs. However, this is not a realistic condition. We therefore focus on the performance of the other two conditions, that set thresholds on when influence propagates.

Indeed, we observe that the coverage achieved by the *EWThr* approach is similar to the baseline one, influencing 80% to 100% of the nodes, while the *AvgThr* approach does not manage to cover the network in its entirety, influencing between 40% to 90% of all the nodes. This is because some nodes in these real-world networks have numerically lower in-degree values while other nodes have numerically higher out-degrees making the edges-weights so small that even an immediate neighbor is unable to influence the adjacent node. On the other hand, the *EWThr* threshold condition depends on structural characteristics of every node, and this might be the reason of its better performance.

Looking more closely on the ranking methodologies, we observe that all are comparable in terms of number of influencers, with *Outdeg* performing slightly better for the AstroPhysics and Yelp subgraphs, and *PR* performing slightly better for the Epinions subgraph. However the differences are minimal (possibly due to the size of the subgraphs) and therefore we cannot draw definite conclusions.

Finally, looking at the performance of the *NewGreedy* algorithm on the AstroPhysics dataset, which is the only common one (the graph was considered undirectional in [5]³), we observe comparable results.

Note that in the implementation of the influence propagation methodology proposed in this paper, the graph is reversed for finding influence of a node so that incoming edges become outgoing edges and vice versa. This is done to capture influence (incoming edge) of a node. By reversing the incoming node, outgoing reach of a particular node can be determined in the entire network. This can be understood by taking an example from the Twitter social network, where a person may follow Obama but Obama does not follow them. In the graph this will be represented as an incoming edge toward Obama. However for the purpose of our algorithm, and in order to find the influence of Obama (and, by extension, of all users) over the entire network, the edge directions should be reversed.

Moreover, the edge-weights are normalized by taking the logarithm of their values. This generates a uniform distribution of edge-weights. The new normalized values of edge-weights

are *negative*, hence to decrease the influence of edge-weights with increasing cascade, the *hopping factor* should be greater than 1 (equation 3), making edge-weights of subsequent hops smaller in numerical value.

B. Social Recommendations

In this experiment we evaluate how influence propagation in the social graph can be leveraged to enhance the accuracy of a recommender system. Using the proposed TB-IP Algorithm described in Section V, we generated neighborhoods of the *influential* users in the Yelp social network, derived from the Yelp Las Vegas dataset⁴.

We designed three social graph-derived datasets to be used as input to the recommender system, by considering the three different strategies of *threshold* selection and optimal number of *hops*. For our baselines we generated datasets starting with the same seed users as the social graph-based ones, and adding equal number of additional users. The difference is that the latter were randomly selected (i.e. the social graph elements were not taken into consideration). In total, we created seven different datasets to be used as input to the recommendation process, resulting in seven different experiments. We should point out that, since the actual users included in each dataset are different, the number of respective ratings and businesses is also different, as shown in Table II. Additionally, for *TwoD NoThr* dataset, we considered only those seed nodes that influenced minimum 5 unique nodes. For both *ThreeD AvgThr* and *TwoD EWThr*, the top-ranked node itself influenced a large portion of other nodes in the network.

TABLE II
DATASET CHARACTERISTICS

Dataset	# Users	# Businesses	# Ratings
TwoD NoThr	146,938	25,611	859,310
BaseD1	146,938	25,028	661,414
ThreeD AvgThr	182,464	25,799	953,237
BaseD2	182,464	25,703	815,350
TwoD EWThr	96,900	24,632	733,408
BaseD3	96,900	23,245	432,714
BaseAll	247,111	26,304	1,104,768

TABLE III
RMSE AND MAE FOR DIFFERENT VERSIONS OF NEIGHBORHOOD GENERATION WITH MOST OPTIMAL PARAMETERS

Experiment	RMSE	MAE
TwoD NoThr	1.33	1.04
BaseD1	1.39	1.11
ThreeD AvgThr	1.34	1.05
BaseD2	1.39	1.10
TwoD EWThr	1.29	1.01
BaseD3	1.40	1.12
BaseAll	1.38	1.09

Dataset 1 - Two hops with no threshold (TwoD NoThr).

In this experiment we considered two hops in the network and no *threshold* for influence propagation for forming neighborhood in the algorithm above. For this we selected 2982 seed

³Results are shown in [5] - Figure 5.

⁴We used the following attributes: *user id*, *friends ids*, *business id*, and *ratings*.

Fig. 2. Experimental Analysis of TB-IP Algorithm on Astro-Physics dataset.

Fig. 3. Experimental Analysis of TB-IP Algorithm on Epinions dataset.

Fig. 4. Experimental Analysis of TB-IP Algorithm on Yelp dataset.

users and expanded it by adding users influenced by these seed users, forming their neighborhood.

Dataset 2 - Baseline Two (BaseD1). This experiment is baseline for Dataset 1 where we have expanded the original dataset of 2982 seed users by randomly selecting other users.

Dataset 3 - Three hops with average threshold (ThreeD AvgThr). In this experiment we considered three hops in the network and the average of all edge-weights as *threshold* condition for considering a node as influenced. For this, we took top-ranked seed user and expanded it by adding users influenced by this user.

Dataset 4 - Baseline Three (BaseD2). This experiment is baseline for Dataset 3 where we have expanded the original dataset of the seed user by randomly selecting other users to make the total number of users equal.

Dataset 5 - Two hops with edge-weight dependent threshold for every node (TwoD EWThr). We set the *threshold* for a node u to be influenced to be equal to the average edge-weight $w(u_v)$ of outgoing edges and consider influence propagation for up to 2 hops. Thus this *threshold* is vertex-dependent. To form the input dataset, we begin by the top-ranked user and expand the neighborhood by adding users influenced by this seed user.

Dataset 6 - Baseline One (BaseD3). We create a baseline dataset for Dataset 5 by randomly selecting an equal number of users (the seed user is included in this set).

Dataset 7 - Baseline All (BaseAll). In this experiment, we used the entire dataset including all Yelp users who have rated at least one business in Las Vegas.

We used the 7 datasets as input to the matrix factoriza-

tion algorithm⁵ and performed 10-cross-fold-validation for all possible combinations of the following parameters: $rank = \{5, 10, 20\}$, $regularization = \{0.1, 0.05, 0.01, 0.5, 0.2\}$, and number of iterations $maxIter = \{10, 20\}$ to evaluate the results.

For this purpose we employed the popular error-based metrics root-mean-square error (RMSE) and mean absolute error (MAE). Assuming that the recommender system generates predicted ratings $\hat{r}_{ui}$ for a test set T of user-item pairs (u, i) for which the true ratings r_{ui} are known. *Root Mean Squared Error (RMSE)* between the predicted and actual ratings is given by:

$$RMSE = \sqrt{\frac{1}{n} \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2} \quad (4)$$

where n is the size of set T . *Mean Absolute Error (MAE)* is a simpler alternative, given by:

$$MAE = \frac{1}{n} \sum_{(u,i) \in T} |\hat{r}_{ui} - r_{ui}| \quad (5)$$

The best (i.e. lowest) RMSE and MAE for each Dataset/Experiment are reported in Table III⁶. We observe that the social graph-based dataset/experiment combos outperform the respective baselines. Moreover, we observe that the vertex-specific threshold strategy (*EWThr*) outperforms the other two. This confirms our assumption that, while the *NoThr* strategy seemed to perform better in terms of coverage/number

⁵We used the ALS Matrix Factorization in pySpark MLlib library.

⁶The best settings were $f=5$ and $\lambda=0.5$ for RMSE, and $f=5$ and $\lambda=0.2$ for MAE.

of influentials in our previous experiments with the small subgraphs, it is a naive approach that will not reflect real-life relationships depicted in large social networks like the one we used here.

Therefore, we verify our initial intuition that the recommendation process greatly benefits when enhanced with social graph data. In addition to that, we observe that the dynamic threshold strategy is the dominating one in this implicit evaluation of the algorithm as well.

VII. CONCLUSION

In this work, influence propagation is considered as a prominent characteristic for information diffusion. We introduce a threshold-bounded influence propagation algorithm to determine this cascading effect. We establish three conditions for determining the threshold for a node to get influenced. Along with this, three approaches are employed for the initial ranking of nodes. These variations are then extensively evaluated by experimental analysis on real-world datasets. The results show that node-dependent threshold conditions are a better choice than global threshold conditions for influence propagation. We subsequently used the proposed algorithm to generate social graph-based neighborhoods. These were used as input to the recommendation algorithm. Our experiments against non-socially enhanced baselines verified our intuition that social recommender systems are indeed more accurate than the traditional rating-based ones.

We are currently exploring additional *threshold* and *influence propagation* strategies to further enhance the recommendation process. We plan to experiment with other centrality measures, such as Katz centrality and are in the process of evaluating the algorithm with larger graphs. We are also exploring ways in which the social data can be integrated in the recommender algorithm itself, instead of being used as a preprocessing step. We are currently working on designing a unified framework that integrates influence propagation and its outcomes, including the neighborhood and the derived social weights in social recommenders as a social regularization term in matrix factorization.

VIII. ACKNOWLEDGMENTS

This work was partially funded by the SJSU Charles W. Davidson College of Engineering student scholar research award.

REFERENCES

- [1] M. Granovetter, *Getting a job: A study of contacts and careers*. University of Chicago press, 2018.
- [2] M. S. Granovetter, "The strength of weak ties," in *Social networks*. Elsevier, 1977, pp. 347–367.
- [3] C. Aggarwal, *Recommender Systems: The Textbook*, 1st ed. Springer Publishing Company, Incorporated, 2016.
- [4] I. Guy, *Social Recommender Systems*. Boston, MA: Springer US, 2015, pp. 511–543. [Online]. Available: https://doi.org/10.1007/978-1-4899-7637-6_15
- [5] M. Eirinaki, N. Moniz, and K. Potika, "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)*, Oct 2016, pp. 408–415.
- [6] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, "Online influence maximization," in *Proc. of KDD'15*, pp. 645–654.
- [7] S. Chen, J. Fan, G. Li, J. Feng, K.-I. Tan, and J. Tang, "Online topic-aware influence maximization," *Proc. VLDB Endow.*, vol. 8, no. 6, pp. 666–677, Feb. 2015.
- [8] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: Finding topic-sensitive influential twitterers," in *Proc. of WSDM'10*, 2010, pp. 261–270.
- [9] D. A. Kim, A. R. Hwang, D. Stafford, D. A. Hughes, A. J. O'Malley, J. H. Fowler, and N. A. Christakis, "Social network targeting to maximise population behaviour change: a cluster randomised controlled trial," *The Lancet*, vol. 386, pp. 145–153, 5 2015.
- [10] M.-E. G. Rossi, F. D. Malliaros, and M. Vazirgiannis, "Spread it good, spread it fast: Identification of influential nodes in social networks," in *Proc. of WWW'15 Companion*. ACM, 2015, pp. 101–102.
- [11] R. Huckfeldt and J. Sprague, "Networks in context: The social flow of political information," *American Political Science Review*, vol. 81, pp. 1197–1216, 12 1987.
- [12] E. Dubois and D. Gaffney, "The multiple facets of influence: Identifying political influentials and opinion leaders on twitter," *American Behavioral Scientist*, vol. 58, no. 10, pp. 1260–1277, 2014.
- [13] Y. Yang, X. Mao, J. Pei, and X. He, "Continuous influence maximization: What discounts should we offer to social network users?" in *Proceedings of the 2016 International Conference on Management of Data*, ser. SIGMOD '16. New York, NY, USA: ACM, 2016, pp. 727–741. [Online]. Available: <http://doi.acm.org/10.1145/2882903.2882961>
- [14] M. Kaple, K. Kulkarni, and K. Potika, "Viral marketing for smart cities: Influencers in social network communities," in *9th IEEE International Workshop on Big Data Applications in Smart City Development*, 2017.
- [15] P. M. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. of KDD'01*, pp. 57–66.
- [16] A. S. Badashian and E. Stroulia, "Measuring user influence in github: The million follower fallacy," in *2016 IEEE/ACM 3rd International Workshop on CrowdSourcing in Software Engineering (CSI-SE)*, May 2016, pp. 15–21.
- [17] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, "Simulated annealing based influence maximization in social networks," in *AAAI*, vol. 11, 2011, pp. 127–132.
- [18] Y. Zeng, X. Chen, G. Cong, S. Qin, J. Tang, and Y. Xiang, "Maximizing influence under influence loss constraint in social networks," *Expert Systems with Applications*, vol. 55, no. Supplement C, pp. 255 – 267, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S095741741600018X>
- [19] M. Gong, J. Yan, B. Shen, L. Ma, and Q. Cai, "Influence maximization in social networks based on discrete particle swarm optimization," *Information Sciences*, vol. 367–368, no. Supplement C, pp. 600 – 614, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002002551630500X>
- [20] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, Oct 1995, pp. 39–43.
- [21] D. Kempe, J. Kleinberg, and Éva Tardos, "Maximizing the spread of influence through a social network," *Theory of Computing*, vol. 11, no. 4, pp. 105–147, 2015.
- [22] S. Hangal, D. Maclean, M. S. Lam, and J. Heer, "All friends are not equal: Using weights in social graphs to improve search."
- [23] J. M. C. Miller McPherson, Lynn Smith-Lovin, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, pp. 415–444, 2001.
- [24] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Netw. Analys. Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.
- [25] I. Guy and D. Carmel, "Social recommender systems," in *Proc. of WWW'11*. ACM, 2011, pp. 283–284.
- [26] J. Tang, H. Gao, X. Hu, and H. Liu, "Exploiting homophily effect for trust prediction," in *Proc. of WSDM '13*. ACM, 2013, pp. 53–62.
- [27] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. of SIGIR '09*. ACM, 2009, pp. 203–210.

- [28] J. Tang, H. Gao, and H. Liu, "mtrust: Discerning multi-faceted trust in a connected world," in *Proc. of WSDM '12*. ACM, 2012, pp. 93–102.
- [29] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. of RecSys '10*. ACM, 2010, pp. 135–142.
- [30] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, in *Proc. of WSDM '11*. ACM, 2011, pp. 287–296.
- [31] P. Massa and P. Avesani, "Controversial users demand local trust metrics: An experimental study on epinions.com community," in *Proc. of AAAI-05*, 2005.
- [32] J. Golbeck, "Trust on the world wide web: a survey," *Found. Trends Web Sci.*, vol. 1, no. 2, pp. 131–197, 2006.
- [33] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. of RecSys '07*. ACM, 2007, pp. 17–24.
- [34] I. Guy, M. Jacovi, E. Shahar, N. Meshulam, V. Soroka, and S. Farrell, "Harvesting with sonar: The value of aggregating social network information," in *Proc. of CHI '08*. ACM, 2008, pp. 1017–1026.
- [35] P. Victor, M. De Cock, and C. Cornelis, "Trust and recommendations," in *Handbook of Recommender Systems*. springer, 2010, pp. 645–675.
- [36] I. Varlamis, M. Eirinaki, and M. Louta, "A study on social network metrics and their application in trust networks," in *Proc. of ASONAM 2010*. IEEE, 2010, pp. 168–175.
- [37] —, "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, 2013, pp. 49–74.
- [38] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL-WP-1999-0120. [Online]. Available: <http://ilpubs.stanford.edu:8090/422/>
- [39] Q. Liu, B. Xiang, N. J. Yuan, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, "An influence propagation view of pagerank," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 3, pp. 30:1–30:30, Mar. 2017. [Online]. Available: <http://doi.acm.org/10.1145/3046941>
- [40] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *International semantic Web conference*. Springer, 2003, pp. 351–368.
- [41] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densefication and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.