**A Probability PPV Model for Social Network Influence Maximization Problem**

Yunjia Ge and Dong Wang  
Shanghai Jiaotong University, Shanghai, China

**Abstract**—For Influence Maximization (IM) problem based on social network, effective and personalized probability learning method was still not theoretical guaranteed. In this paper, we proposed a PPV probability model based on IM problem, which effectively learnt influence probabilities and personal-ized influence for each node pair. By clustering user groups and analyzing similarity of users from both offline action log and online social network topological structure, we estimated reliable parameters of our probability model, differed user’s influence with different features on its neighbors, and improved probability learning procedure. PageRank algorithm and fuzzy cognitive map concept help to validate our model in PPV calculation. Experiments show that our approach outperforms the state-of-the-art algorithm. As preference property $\gamma$ and transition property $p$, which is called PPV in our model, explains features of influence between user pairs, accuracy of personalized influence probability is improved.

**Keywords**—influence maximization; viral marketing; PageRank algorithm; personalization

I. INTRODUCTION

Widely use of smart devices promotes the popularity of online social network. One of the derivatives, viral marketing based on relationship online, has been adopted by most companies instead of traditional advertisement.

When a novel idea or product sprouts in market, its survival rate depends on how much people accept it - how people affect others and thus how the idea or product information spreads via social network. Researches [1] show that people trust the information coming from their friends in close social circle, rather than general channels like newspapers or advertisements. Formalized definition of this problem is “Influence Maximization (IM)”, aiming to find top K seed nodes which activates other nodes quickly (over a fixed threshold value). Companies find the viral marketing useful for market purposes, and has awaken an interest in research over recent years.

IM was first proposed by Domingos and Richardson [2], based on which Kempe et al. [3] formulated the problem: Given a network graph and threshold number k, find the set of top-k influential seed nodes, and marketing efforts can then be focused on them. Kempe verified that the problem is NP-hard, and the greedy algorithm has a constant approximation guarantee. Large storage space and time cost is still required by the approximation algorithm due to the explosive growth of nodes.

Nodes are affecting each other via a direct influence graph $G$, which is essentially a graph with influence-probability-weighted edges and nodes representing social network users[2,4,5,6,]. For example, edge (C, E) with weight 0.5 means that user E may be influenced by user C with a probability of 0.5.

However, most existing IM algorithms focuses on propagation model or approximation algorithm based on a
complete IM graph G, assuming that the influence probability between each user is already known. This assumption may not hold. Consider such a situation: products need to be popularized in a new city, where company has no idea how people there influence others, thus the firms just cannot run IM algorithms to find top k seed nodes to active.

Goyal et al. [9] have proposed a solution based on real data learning, which simulates the probability between users. The result is by far much more accurate than general probability assignment. Literature studying this topic relies exclusively on online social network, such as Facebook, Twitter, etc., to look for user relationship and study influence probabilities. However, there are still two problems: 1) ignore different impacts on different fields of the same seed user. E.g., A professor has a much more convincing impact on his social circle in education field, rather than other unprofessional one. 2) extract probability only from online data, thus only consider the virtual connections rather than physical one.

To tackle this problem, we propose a PPV (Personalized PageRank Vector) probability model to simulate user relationship with both online social network and offline action log. Figure I illustrates our model framework, which reaches the following contributions:

1) Learn influence probabilities, and personalize influence for each node pair and typical event.

2) Construct probability learning model with sufficient theoretical foundation, E.g. matrix constrin-gency. Take social graph as input of IM problem and output top k influential seed nodes.

3) As the time cost and space explodes with the growth of nodes, pruning method and Monte Carlo are introduced to improve the computational efficiency.

II. BACKGROUND

A. Influence Maximization(IM) Propagation Model

IM based on social network was first proposed by Domingos and Richardson [2]. Kempe et al. [3] formulated IM problem, verified that finding top k seed nodes is NP-hard, and focused on two types of propagation model to simulate the spread of influence through network. Once all the nodes and edge weights are defined, all active nodes impact their neighbors in every time step.

In the independent cascade (IC) model, for example, a graph edge e(v,w) from user v to user w with probability weight p implies that user v has a chance p to activate its inactive neighbor w. While in the linear threshold (LT) model, as long as accumulative effect of w from neighbors reaches its activated threshold value, w turns into an active node and impact others in the next time step. The state transition of a node is irreversible, thus different choice of seed nodes may lead to different result set of active nodes in the final time step. Given a direct influence graph G, IM problem aims to maximize the size of activated node result set, or called influence, with a seed node number limit k.

While several heuristics like Degree Discount[5] and IRIE[12] were developed to efficiently find IM solutions, their accuracy were not theoretical guaranteed. Improved approximation algorithms include CELF [13] and CELF++ [11]. Recently, Borgs[14] showed his reverse influence sampling runtime-optimal with accuracy guarantees.

B. Influence Probabilities Learning

Rather than traditional ad-hoc assumptions, Goyal et al. [9] proposed a static model to learn influence probabilities between users from propagation trace data. Extracting influence from real online social networks service has been proved effective, several experiment results [10,11] demonstrated the validity of this approach. However, as the social network is unstable and contains thousands of nodes, large storage space and time cost is still required by necessity. What's more, existing practices have not differentiated influence on different features of the same user. Our framework personalizes user influence on his social circle with data mining methods, such as PageRank algorithm, delves into users and propagated item/issue for features behind the spread of social network influence.

C. PageRank

PageRank, as a link analysis algorithm, was first proposed [16] to calculate webpage importance for query in World Wide Web, and used by Google search engine for years. The heuristic underlying PageRank is that importance of a page, also ranking, is proportional to its parents' importance, and inversely proportional to outgoing degree L(). In general, the PageRank value for any page \( P_i \) can be expressed as:

\[
PR(P_i) = \sum_{j \in L_i} \frac{PR(P_j)}{L(P_j)}
\]

(1)

\( E_p \) is the set containing all pages linking to page \( P_i \). Similar methods such as classic HITS [15], hierarchical rank [17] differ widely in ranking calculating, but all depend on the link structure and ignore the contents of query.

A page without outgoing links is called a sink, and therefore terminates the ranking calculating as L(p)=0. To tackle the problem, pages without outgoing links are generally assumed to link out to all other pages, thus the PageRank value can be expressed as:

\[
PR(P_i) = \frac{1 - d}{N} + d \cdot \sum_{p \in \text{out}(i)} \frac{PR(P_j)}{L(P_j)}
\]

(2)

where d is a residual probability for stationary distribution, usually set d to 0.85 (by Google).
Algorithm 1 depicts the solution of PageRank iterate problem. Given a direct graph \( G = (V, E) \), edge \((P_i, P_j)\) represents there is a link from page \( P_i \) to page \( P_j \). Matrix \( P \), which implies rankings of all the pages, is successive updated during each iteration. Once \( L_1 \) norm of difference value vector \( |P_k - P_0| \) is less than default value \( \varepsilon \), PageRank iteration is terminated.

D. Topic-Sensitive PageRank

Based on PageRank algorithm and its link analysis, topic-sensitive PageRank provides a scalable approach for personalizing search rankings. Chakrabarti et al. [18], Richardson et al. [19] demonstrate that search rankings are sensitive to topics of page. To exploit search context in a tractable and effective way, Haveliwala [20] assigned multiple a-priori importance estimates to pages. Each score provided an importance estimate with respect to topics. These scores constituted topic’s probability distribution of weights over basic topics selected from Open Directory. The concept of resolving query into topic probability distribution is similar to one of our model’s goals, to personalize influence for each node pair and typical event.

III. PPV Probability Model

As mentioned above, a professor generally has a much more convincing impact on his social circle in education field, rather than other unprofessional ones. However, the difference on different fields impact of the same seed user are often ignored. Based on social network and influence maximization problem, our framework aims to build a Personalized PageRank Vector (PPV) Probability Model, aiming to personalize influence for each node pair and typical event.

Matrix factorization, as a data-mining basic method to personalize user’s influence, helps to analysis similar features shared between infecting user and infected user, between infecting user and spread event, and between infected user and spread event. Our PPV probability model based on PageRank algorithm extracts these features from both online social network topological structure and offline user action log, then simulates a stationary probability distribution induced by Monte Carlo random walk on huge social network.

A. Review of IM Problem

First let us review IM Problem probability learning: the essential idea is that if user \( u \) has a social link to user \( v \), then the action taken by user \( u \) implicitly confer some influence to user \( v \). How much influence does \( u \) confer? To simulate such situation and calculate the influence based on user features and action feature, let us review PageRank algorithm. As for PageRank, the essential idea is that if page \( u \) has a link to page \( v \) in WWW, the author of \( u \) implicitly confer some importance to page \( v \).

### Table I. Pagerank and IM Probability Learning

<table>
<thead>
<tr>
<th>Edge (Weight)</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM Problem</td>
<td>Influence (weight: influence probability)</td>
</tr>
<tr>
<td>PageRank Algorithm</td>
<td>Link (weight: transition probability)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IM Problem</th>
<th>Threshold Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Influence on its neighbors (E.g. after A take action X)</td>
<td>Probability of its neighbor B take the same action X</td>
</tr>
<tr>
<td>Attributes of Users</td>
<td>Feature of Query</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IM Problem</th>
<th>Probability of transition (E.g. after choosing A, probability of choosing B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank Algorithm</td>
<td>Probability of transition (E.g. after choosing A, probability of choosing B)</td>
</tr>
</tbody>
</table>

Table I shows the details of similarities between PageRank and IM probability learning: 1) Topological structures are similar, both of which are based on directed social graph. In IM problem, a graph edge \((e,v,w)\) from user \( u \) to user \( w \) with probability weight \( p \) implies that user \( u \) has a chance \( p \) to activate its inactive neighbor \( w \). While in PageRank, given a graph \( G \), edge \((v,w)\) represents there is a link from page \( v \) to page \( w \). 2) Conditions of state transition are similar. For IM problem, users are effected by close social circle to purchase similar products, also called viral marketing. While for PageRank, pages are selected according to their importance ranking, which is topic-sensitive and determined by attribute-distribution of query.

Based on theoretical similarity between IM probability learning and PageRank algorithm, our model analysis the motivation behind IM problem again and construct Personalized PageRank Vector (PPV) for each node to represent the features behind user.

B. Probability Model Definition

Let \( L_u \) be outgoing degree of user \( u \), and \( P^{(k)}_v \) represent the influence of user \( u \) at time step \( k \). Generally the link \((u,v)\) confers \( P^{(k)}_v \) units of importance to \( v \). Assume there are \( N \) nodes in the whole social graph, assign all users the initial influence value \( 1/N \). In every time step, influence probability of each node is expressed as follows by analogy with PageRank algorithm:

\[
\forall v \quad P^{(k+1)}_v = \sum_{u,\text{link}} \frac{P^{(k)}_u}{L(u)}
\]  

(3)

To enforce this assumption, for users with no outgoing link, like sink in PageRank algorithm, we can either iteratively remove such nodes in direct influence graph \( G \), or conceptually
add a complete set of outgoing edges to these sink nodes. Both of them are guaranteed feasible in researches, while in IM problem we prefer the latter one for marketing purpose - firms are always hoping to mine potential users as many as possible. Let \( P \) represents new transition matrix with new-added transition edges with probability \((1-d)/N\) between node pairs in direct influence graph:

\[
\tilde{P}^{(k+1)} = dP + (1-d)(1/n) \cdot E_{N \times N}
\]

(4)

Let \( \varepsilon = 1-d \), vectors in new matrix \( \Pi^{(k)} \) represents influence probability of node pairs, which is also influence transition probability in \( \tilde{P} \). model of IM probability learning, which is called PPV in our model. The distribution of random walk \( \varepsilon \) with probability \( \varepsilon \). The distribution of random walk termination is subject to \( \pi \) distribution, which simulates probability that the terminal will be chosen.

D. PPV Calculation

Formula (7) can be simplified with preference property \( \gamma \) and transition property \( p \), which is called PPV in our probability model standing for the bond between node pairs.

\[
\Pi = (1-\varepsilon) \cdot \gamma + \varepsilon \cdot (A \cdot p)
\]

(9)

For example, given viral marketing background and a direct social graph \( G \), \( p = PPV(i,j) \) represents user \( u \), who have already purchase \( X \), has a probability \( p \) to tempt his social friend \( v \) to purchase the same product. PPV \( (i,j) \) depends on the similarity of user \( u \) and user \( v \), and also rest with features of item \( X \). In our probability model, PPV \((i,j)\) is extracted from user's action log to estimate influence. User action is divided into two categories: First-behavior action and Group-influence Action.

Algorithm 2 Discover user groups

| INPUT: Action set AS(User, Time, ASid), \( \delta \) as threshold |
| OUTPUT: User Group set AC |
| 1: Set AliveCluster to empty |
| 2: for AS(\( i \)) in AS do |
| 3: Intercact \( i \) as AS(\( i \)) to AC by User -> CurrentCluster |
| 4: for \( \varepsilon \) in CurrentCluster |
| 5: if \( \varepsilon \)'s size is less than 2 then break; |
| 6: if AC \( \gamma \) = \( \gamma \) then frequency(AC) + 1; |
| 7: else Add \( \delta \) to AliveCluster |
| 10: End for |
| 11: Add AS(\( i \)) into AliveCluster with frequency(ask)=1 if AS(\( i \)) not in |
| 12: Output AC(\( i \)) in AC where frequency(AC) >= \( \delta \) |
| 13: Remove AC(\( i \)) |
| 14: End for |

First-behavior action: the very first record in user action log, considered mainly effected by the degree of accepting and purchasing new things of user, which is relatively stable. With first-behavior actions, our model gets a binary fuzzy influence map. Take viral marketing as an example. Assuming that there exists a link from user \( u \) to user \( v \) in social graph topological structure. User \( u \) purchase product \( X \) at \( t_1 \), and user \( v \) purchase the same product at \( t_2 \). If \( 0 < t_2 - t_1 \leq \Delta t \), user \( v \) is considered influenced by user \( u \), and PPV \((u,v)\) is considered 1. \( \Delta t \) is threshold time interval decided by action itself, generally uniform. Otherwise PPV \((u,v)\) is considered 0 as the influence is unknown.
Group-influence Action: users in the same user group are effected by each other to some extent. The influence within group is initiated with Euclidean Distance of time context and spatial context. PPV(u,v) then is optimized by group influence and decay with time. How to recognize user group? Take purchasing action as an example, users purchase products in the same item cluster within certain time interval are considered one user group. Detailed steps are shown in Algorithm 2.

The probability learning of PPV(u,v) can be considered as an optimization from fuzzy cognitive map to probability cognitive map. It is a procedure for modifying its weight matrix in order to train the PPV to model the behavior of IM system. The whole PPV model outputs influence probability between all node pairs as matrix Π.

### IV. EXPERIMENT EVALUATION

To learn influence probability of user pairs in general influence problem, we performed experiments to demonstrate the accuracy and efficiency of the proposed approach against traditional ad-hoc (AD) assignment and simple action log extraction (SALE) approaches. Experiments are conducted on Gowalla check-in datasets from Feb. 2009 to Oct. 2010, which contain location-based check-in data as action logs of 196,591 users.

Our simulator first uses dataset to obtain a graph G, then initially associate an influence probability to each edge in G with \( \pi_{uv} = 1 / L_{uv} \). After action log analysis with PPV calculation algorithm, the simulating influence probability matrix \( \Pi \) is updated with iteration until L1 norm of difference value vector is less than default value \( \epsilon = 0.0005 \). Probabilities are learnt and refined during these trials. We set the residual probability as 0.85 in our PPV model and validate probability learning algorithm with training-test. All the experiments were conducted under entity framework 6.0 on a Windows server with 4G memory.

After cleaning dirty data, there are still two steps before calculating PPV. 1) Matrix Smoothing: remove spider trap and dead end iteratively in order to ensure the probability distribution is stationary.

2) User Group Clustering: select threshold \( \delta_q \) in 2 values and recognize user group for PPV calculation (21081 groups with 0.0005°, and 42651 groups with 0.001°).

<table>
<thead>
<tr>
<th>Threshold δq</th>
<th>Metrics</th>
<th>AD assignment</th>
<th>SALE learning</th>
<th>PPV learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0005</td>
<td>MAE</td>
<td>1.885</td>
<td>0.917</td>
<td>0.702</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>2.269</td>
<td>1.239</td>
<td>0.862</td>
</tr>
<tr>
<td>0.001</td>
<td>MAE</td>
<td>1.510</td>
<td>0.720</td>
<td>0.608</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>1.834</td>
<td>0.955</td>
<td>0.803</td>
</tr>
</tbody>
</table>

We randomly select 80% of data as training set the remaining 20% as validation set, conduct five times and take the mean as final result. The simulation compares MAE and RMSE with methods mentioned above. Table II shows the result that PPV influence probability learning is more effective in fine grit, better than SALE learning and much better than traditional AD assignment.

### V. CONCLUSION

In this paper, we have proposed a PPV probability model based on IM problem which effectively learn influence probabilities in social network, and personalize influence for each node pair. By clustering user groups and analyzing similarity of users from both offline action log and online social network topological structure, we estimate reliable parameters of our probability model, differ user's influence with different features on its neighbors, and improve probability learning procedure. PageRank algorithm and fuzzy cognitive map concept help to validate our model.

Experiments show that our approach outperforms the state-of-the-art algorithm in the following aspects: 1) As preference property \( \gamma \) and transition property \( p \), which is called PPV in our probability model, explains features of influence between user pairs, the accuracy of influence probability is improved. 2) Influence probability between all user pairs are learnt in final steady distribution. Advanced personalized recommendation can be extended.

Our approach is still quite general, approach to extracting different features of influence can be extended to contextual information. We plan to take context information into consideration so that similarity between users and similarity between user and item that is spread through social network can be more accuracy.

### REFERENCES


