This paper was presented as part of the main technical program at IEEE INFOCOM 2011

Relationship Classification in Large Scale Online Social Networks and Its Impact on Information Propagation

Shaojie Tang[*] Jing Yuan[♭] Xufei Mao[§] Xiang-Yang Li[*] Wei Chen[†] Guojun Dai[‡]
[*]Department of Computer Science, Illinois Institute of Technology
[♭]Department of Computer Science, NanJing University [†]Microsoft Research Asia
[§]Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia
Beijing University of Posts and Telecommunications
[‡]Institute of Computer Application Technology, Hangzhou Dianzi University

Abstract—**In this paper, we study two tightly coupled topics in online social networks (OSN): relationship classification and information propagation. The links in a social network often reflect social relationships among users. In this work, we first investigate identifying the relationships among social network users based on certain social network property and limited pre-known information. Social networks have been widely used for online marketing. A critical step is the propagation maximization by choosing a small set of seeds for marketing. Based on the social relationships learned in the first step, we show how to exploit these relationships to maximize the marketing efficacy. We evaluate our approach on large scale real-world data from Renren network, showing that the performances of our relationship classification and propagation maximization algorithm are pretty good in practice.**

I. Introduction

Online social networks (including sites like Youtube, Orkut, Renren and messengers like Skye and MSN) are among the most popular sites and communication tools on the Internet. The users of these sites and tools form huge social networks. Using online social networking, you can influence others and also leverage the power of others' influence. When we focus on the right relationships and the right people, social networking provides us tremendous leveraging power - one that creates word of mouth and buzz marketing at levels differing from other marketing strategies. It motivates the research community to conduct extensive studies on the influence maximization problem under various settings. Consider the following example. A video game company develops a new video game and tries to promote it through an online social network. Under a limited budget, the company can only distribute limited number of free samples to a few initial users. Hopefully, those users would enjoy the game and promote it to their friends. The problem is whom to select as the initial users to maximize the influence. When studying the influence maximization problem, by far the biggest mistake may be focusing on the quantity of connections, not the quality. The intuition, we guess, is that the bigger the Net is the more contacts will happen, as a result, the larger influence we will achieve. Unfortunately, this is usually not true in real world. The point is if our audience is not actively adopting or helping you spread our ideas, we are not able to achieve large influence. Most previous works assume the relationship between every people is the same. Possibly true, but it is wiser to just target at the right audience and right relationship that provide you with more opportunities to influence them. This is one of the most important motivations why we study the relationship classification and influence maximization together in this paper.

In the previous example, it may be ineffective to run through Facebook groups with largest number of members or edges and then distribute their samples for online marketing. A better way could be to classify all (or majority) links in the social network according to their relationship, find a set of seeds such that the influence can be maximized through the links with desired relationship, *e.g.*, those users who may have common interest in video games. In summary, a typical social network likely contains multiple relations, and different relations have different importance in reflecting the user's information need. In the traditional social network analysis, people do not distinguish these relations and the different relations are equally treated. So, they are simply combined together for describing the structure between objects. The main contributions of this paper can be summarized as follows:

- As one of the first work addressing **relationship classification problem**, we try to classify the relationships of different social network links based on a small subset of known social relationships and certain social network property.
- By taking different relationships into account, we further investigate the impact of relationship classification on **information propagation problem**. Leveraging the social relationships learned in the first step, we show how to exploit these relationships to maximize the information propagation.
- We extensively evaluate our algorithms on a large scale online social network. To this extent, we have collected more than $60,000$ users' data from Renren network for our evaluation.

The rest of the paper is organized as follows. Section II presents the network model and problem to be studied.

Section III and Section IV study the relationship classification and influence maximization problem. We conduct extensive experiments and report our results in Section V. We briefly review related results in Section VI and conclude the paper in Section VII.

II. Preliminaries

In this section, we will present the social network models used and formally define the problems to be studied in this paper.

A. Social Network Model

An online social network is often composed of users, links, and groups. As in all online social networks, to participate fully in an online social network, a user (often a human being) must register with the site. The user profile collected by the site contains the volunteered information about the users, which could be bogus sometimes. After a user registered in a site, the user can create links to other users in the same social network. Here users form links for various reasons: the users can be real-world acquaintances, or business contacts; they can share some common interests; or they are interested in each other's contents. For a user u, the set of users with whom u has links are called the *contacts* of the user u.

Popular online social networking sites such as Flickr, Renren and Orkut, rely on an explicit user graph to organize, locate, and share content as well as contacts. For most online social networks (such as Renren net), a user's contacts and his/her profile are often visible to those users who visit the user's account. Some sites (such as LinkedIn) only allow users to view the information (contacts and profile) of its contacts. Most sites enable users to create and join special interest groups. Users can post messages to groups and upload shared content to the group. In many of these sites, links between users are public and can be crawled automatically to capture and study a large fraction of the connected user graph.

In many online messenger softwares and online social networking sites, the contacts of a user are classified into different categories. The categories can be fixed (as in LinkedIn) or can be defined by users (as in almost all online messenger softwares like Skye, MSN messenger).

In this paper, we formulate a social network as a graph $G = (V, E)$, in which V is the set of users in the social network, and E is the set of links among users. We assume that the links in the social network can be classified into a set of categories $\mathcal{L} = \{\ell_0, \ell_1, \ell_2, \cdots, \ell_k\}$, such as {friends, classmates, officemates, family, others }. In other words, each link $e \in E$ has one of multiple labels $\ell(e) \in \mathcal{L}$. Note that our scheme can also work when each link holds multiple labelings.

B. Problems To Be Studied

We will focus on the following two closely related problems.

1) Relationship Classification: An explicit assumption made in many existing works on social network mining is that the social network G typically contains only one relation. That is, the relationships among different users are identical as long as they have a link between each other. For example, all links in the network can be regarded as "acquaintance". In other words, all links have one common generic labeling "acquaintance". However, this may not be the case in many real online social networks (such as Renren network, MSN network). If we dig more into relationships among users, we may have multiple possibilities, such as "college classmates", "high-school classmates", "officemates" and so forth. The relationship classification is actually to solve the following question.

Question 1 (Relationship Classification): We are given a social network $G = (V, E)$, a labeling $\ell()$ for a small subset $K \subset E$ of edges, and the set of all possible labels C that can be used. Here a labeling $\ell(e)$ of an edge $e = (u, v)$ denotes that the relation between u and v is of type $\ell(e)$. Our objective is to assign each link e in E a label from C such that the accuracy is maximized. Here we define the accuracy as the ratio of the number of links which are labeled correctly to the total number of unlabeled links.

2) Maximize Product Information Propagation: A social network plays a fundamental role as a medium for the spread of information, ideas, and influence among its members. An idea or innovation will appear in social networks. It can either die out quickly or make significant inroads into the population. If we want to understand the extent to which such ideas are adopted, it is important to understand how the dynamics of adoption are likely to unfold within the underlying social network: the extent to which people are likely to be affected by decisions of their friends and colleagues, or the extent to which *word-of-mouth* effects will take hold. Such network diffusion processes have a long history of study in the social sciences.

In this work, we consider the issue of choosing influential sets of individuals as a problem in discrete optimization. In particular, we assume each user has certain cost at which it can be chosen as an initial active user, further, different nodes may have different weights which reflect their importances of being reached.

Question 2 (product information propagation): We are interested at finding a set of initial active users under the budget constraint such that the total weight of the users they can affect is maximized.

Most importantly, we also study the impact of different relationships on the product information propagation process. In this paper, we mainly study the product information propagation problem under *Independent Cascade Model*(ICM), which is one of the most widely used propagation model. The details about ICM will be explained later.

III. Relationship Classification

In this section, we will study how to classify the relationship in online social networks efficiently and effectively. Due to various reasons, *e.g.* information missing, privacy or security, we may not know the detailed relationship information among all users. Instead, only partial users would like to open their friend list with detailed relationship information. We will mainly focus on learning the social relationships of pairs of nodes in the social network. The problem of relationship classification can be simply stated as follows: In an online social network, based on limited known relationship information, how to predict the relationship of the other links? Specially, we aim at finding a best way to label the links such that some special properties of online social networks can be well obeyed while best approximating the pre-labeled samples.

To the best of our knowledge, this is one of the first work addressing *relationship classification problem* (RCP) in large scale online social networks. Compared with arbitrary network, an online social network often has some specific properties. Recent research on networks among mathematicians and physicists has focused on a number of distinctive statistical properties that most networks seem to share. In this paper, we highlight the following two properties which can be found in many online social networks.

▷ The first one that many networks have in common is *network transitivity*, which is the property that two vertices that are both neighbors of the same third vertex have a heightened probability of also being neighbors of one another. In the language of social networks, two of your friends will have a greater probability of knowing one another than will two people chosen at random from the population, on account of their common acquaintance with you. The probability that two of one's friends are friends themselves typical values in the range of 0.1 to 0.5 in many real-world online social networks.

▷ The second important property is the *community structure*. (This property is also sometimes called cluster.) Consider for a moment the case of social networks of friendships or other acquaintances between individuals. It is a matter of common experience that such networks seem to have communities in them: subsets of vertices within which vertex-vertex connections are dense, but between which connections are less dense.

The above two properties have already been noticed and addressed in a number of literatures. However, in order to solve our relationship classification problem, we need deeper insight into the structure of online social networks. Based on our online data crawled from Renren net, we gain very important observations as follows:

▶ For a Renren user v, if the links between v and two of its acquaintance neighbors u, w have the same labeling (relationship), *e.g.* $\ell(uv) = \ell(wv) =$ "college classmates", then the probability that u and w are also acquaintance neighbors themselves with the same labeling in Renren network is very high (the average probability is around 85%).

▶ We further observe that, for any user v, if the links between v and two of its acquaintance neighbors u, w have different labelings, *e.g.* $\ell(vu) =$ "classmates" and $\ell(vw) =$ "family members", then the probability that u and w are acquaintance neighbors themselves in Renren network is very low (around 0.01 in our survey).

Not only hold for Renren networks, indeed, the above two properties can also be found in a number of other online social networks, we next give an illustration using the following example.

By searching Ming Yao on *http://renlifang.msra.cn* [1], we get a social network centered at Yao which is shown in Fig 1. Yao is one of the most famous Chinese Basketball players, and nowadays he is playing for Houston's basketball team in NBA. Here we add a link between any two individuals if and only if both their names appeared in the same online news. For simplicity, we are interested in classifying all the links into two sets according to their labelings: CBA Teammate and NBA Teammate.

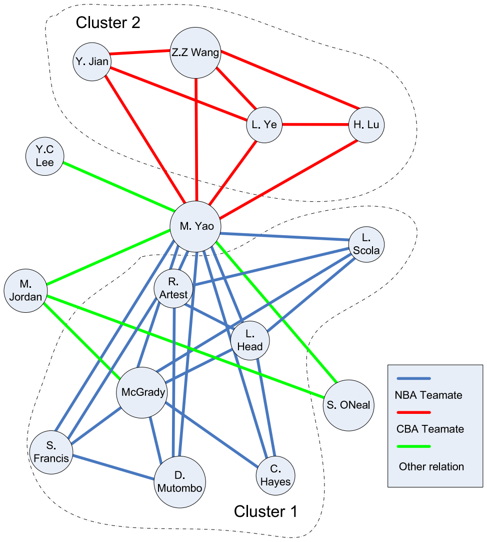


Fig. 1. Illustration of those two observations. If we remove Yao from the social network centered at him, two disjoint "clusters" (Cluster 1 and Cluster 2) have naturally formed across the remaining network, within each cluster each individual tends to have the same relationship among each other.

If we remove Yao from the social network, two disjoint "clusters" (Cluster 1 and Cluster 2) have naturally formed across the remaining network which is shown in Figure 1. If we dig more into the relations among individuals, we find that the way those two clusters are formed is tightly related to the relationship between Yao and them. For example, we observe that the relationship between Yao and McGrady, Battier and Francis *et al.* are all NBA teammate, and McGrady, Battier and Francis *et al.* belong to the same cluster, this structure illustrates our first observation. Further, the neighbors from different clusters more likely have different relations with Yao, *e.g.* Cluster 1 is mainly composed by Yao's NBA teammates and Cluster 2 is formed by his CBA teammates. This illustrates our second observation.

To further confirm the above two observations in large scale online social networks, we take set of 60,000 users from Renren to evaluate. We first define a function $Q_S = N_{inner}/N_{all}$ for any subset S of users, where N_{inner} denotes the number

[1] a social network search engine developed by MSRA.

User ID	Q_H	Q_C	Q_O
1103	0.79	0.77	0.71
1004	0.80	0.76	0.74
1011	0.81	0.75	0.78
....	...	...	...
19999	0.84	0.79	0.77
20000	0.82	0.76	0.76
20001	0.75	0.74	0.73
....	...	...	...
46398	0.81	0.76	0.73
46399	0.78	0.80	0.78
46400	0.82	0.71	0.76
....	...	...	...

TABLE I
IN THIS TABLE, $Q_C(i)$ DENOTES THE Q VALUE OF THE SUBSET FORMED BY ALL v_i'S COLLEGE CLASSMATES. SIMILARLY, $Q_H(i)$ AND $Q_O(i)$ DENOTE THE Q VALUE OF THE SUBSET FORMED BY HIS/HER HIGHSCHOOL CLASSMATES AND OFFICEMATES RESPECTIVELY.

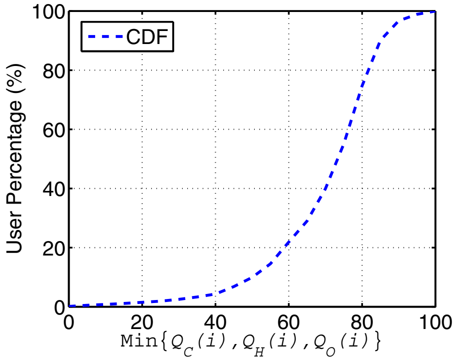


Fig. 2. Cumulative distribution function (CDF) of the value $\min\{Q_C(i), Q_H(i), Q_O(i)\}$.

of edges with both vertices within set S, N_{all} is the number of edges with one or both vertices in set S. Therefore, the Q value measures the fraction of edges falling within community. A larger Q value means stronger community structures. In the following contents we will use Q value to evaluate the quality of the estimated community structure. For each user v_i, let $Q_C(i)$ denote the Q value of the subset formed by all v_i's College classmates for a user v_i. Similarly, let $Q_H(i)$ and $Q_O(i)$ denote the Q value of the subset formed by his/her Highschool classmates and Officemates respectively. In Table. 1, we list the Q value collected from 9 Renren users as a sampling. The table shows that among these 9 online users, the minimum $Q_C(i)$ is at least 75%, the minimum $Q_H(i)$ is at least 71% and the minimum $Q_O(i)$ is at least 71%. And the average value of all the three features is around 75%. Figure 2 depicts the Cumulative distribution function (CDF) of the value $\min\{Q_C(i), Q_H(i), Q_O(i)\}$ for each user v_i. The resulted curve shows that among the $60,000$ users, at least 80% users has the value $\min\{Q_C(i), Q_H(i), Q_O(i)\}$ no less than 60%, and almost half users have the value at least 80%. Note only less than 10% users have the value below 40%. Indeed, both above results confirm that the previous two observations hold for some large scale online social network such as Renren net.

We next leverage these two properties to design our Relationship Classification Algorithm (RCA).

A. *Relationship Classification Algorithm (RCA)*

Given an online social network $G = (V, E)$, we use $G(V')$ to denote the subgraph induced by set of vertices $V' \subseteq V$, $N(v)$ to denote the set of v's neighbors in G. Throughout this paper let $|\mathcal{S}|$ denote the size of set $\mathcal{S}$. We first describe the main idea of our relationship classification algorithm (RCA). As discussed in the previous section, we have a number of important insights into the structure of online social networks. In order to take advantage of those observations to solve our problem, we summarize them in a more formal way as follows.

For a user $v \in V$, the induced subgraph $G(N(v))$ tends to form disjoint "communities" or "clusters" such that 1) The users having the same relationship, say ℓ_i, with v tend to belong to a same cluster in $G(N(v))$, further, they tend to have the same relationship ℓ_i among each other; 2) The users having different relationships with v are more likely present in different clusters in $G(N(v))$.

We next aim at finding a way to label all the links such that the pre-labeled information can be well approximated and the above two properties are maintained. For user v, let $V_i \subseteq N(v)$ denote the set of users having the same relationship ℓ_i with v according to the pre-labeled links. Recall that the users having the same relationship, *e.g.* ℓ_i, with v are more likely to form a cluster in $G(N(v))$. Thus if we are able to find out the set of users belonging to the same cluster formed by V_i, we may claim that all those users more likely will have the same relationship ℓ_i with v.

In order to find out those users belonging to the same cluster as V_i, we run *random walk* starting from each user in V_i to evaluate the "affecting range" caused by V_i. Here we use the probability that at least one of those random walks visits a certain user within fixed steps to measure the affect from V_i on this user. Basically, the more "heavily" one user $w \in N(v)$ is affected by V_i, the more possible it belongs to the same cluster formed by V_i. Then we can assign label ℓ_i to link wv, *e.g.* $\ell(wv) = \ell_i$, with high confidence. Our experiment results show that this scheme works pretty well in real large scale social networks. However, if we process $G(v \bigcup N(v))$ for each vertex $v \in V$ locally independently, we may not be able to guarantee the consistency among the results from different users. For example, when the edge (u, v) is assigned with different labelings after we process $G(v \bigcup N(v))$ and $G(u \bigcup N(u))$ independently, how can we determine the final labeling? Furthermore, when we process $G(v \bigcup N(v))$ independently, due to the limited number of pre-labeled links in $G(v \bigcup N(v))$, we may not have enough information to label all the other links accurately, in contrast, some of v's neighbors may have useful information to help v to do labeling.

In order to tackle those issues, we create a *shared information table* (SIT) to keep the latest labeling information for each link. Each adjacent user is able to access and update the information of the corresponding link stored in SIT. When we process each $G(v \bigcup N(v))$, we must utilize the data stored in SIT to make sure that the information shared among each other is up to date. After $G(v \bigcup N(v))$ is processed, we also need to update the corresponding entries in SIT on time.

B. RCA Design

We now describe RCA in details. Given a social network G, let M_G be the normalized transition matrix of graph G, *i.e.*

$$M_G(v,u) = \begin{cases} 1/|N_G(v)| & \text{if u is v's neighbor in G} \\ 0 & \text{otherwise.} \end{cases}$$

Recall that $|N_G(v)|$ is the number of neighbors of v in G. For a vector $(x_1, x_2, \cdots, x_n)$, we further define a matrix

$$(x_1, x_2, \cdots, x_n)_Z = \begin{pmatrix} x_1 & 0 & .. & .. & .. & 0 \\ 0 & x_2 & 0 & .. & .. & 0 \\ 0 & 0 & .. & 0 & .. & 0 \\ 0 & .. & 0 & .. & 0 & 0 \\ 0 & .. & .. & 0 & x_{n-1} & 0 \\ 0 & .. & .. & .. & 0 & x_n \end{pmatrix}$$

Let $\mathcal{T}$ denote the shared information table (SIT) that maintains the latest labeling information for each link. It has k columns and $|E|$ rows where k is the number of total possible labelings, where the entry located at the i-th column and j-th row, $\mathcal{T}(\ell_i, e_j)$, stores the confidence score by labeling e_j using ℓ_i. Notice that the value of confidence score belongs to $[0, 1]$. In the initial phase, we set

$$\mathcal{T}(\ell_i, e_j) = \begin{cases} 1 & \text{if } e_j \text{ is pre-labeled by } \ell_i \\ 0 & \text{otherwise.} \end{cases}$$

For a subset $U \subseteq \{1, \cdots, n\}$ we will denote by 1_U the 0/1 vector, whose i-th entry is 1 if and only if $i \in U$. Let I_n denote the identity matrix of size n, which is the n-by-n matrix in which all the elements on the main diagonal are equal to 1 and all other elements are equal to 0.

We say a node v is t-affected by set of nodes U with value p if the probability that at least one random walk starting from U can visit v within t steps is p. Here t is a system parameter which will be discussed later. The value p is called the *t-affect* of U on vertex v. The total t-affect of U is the sum of t-affects of U on all vertices V.

Theorem 1: Given an online social network $G = (V, E)$, assume $|V| = n$, $U \subseteq V$ and each vertex $v \in U$ is assigned a unique index $D(v)$ from $\{1, \cdots, n\}$, then after t steps, the total t-affect caused by U is at least:

$$F_G(U, t) = 1_V \times \left[I_n - \prod_{i=0}^{t} \left(1_V - 1_U M_G^i\right)_Z \right]$$

where the i-th entry of $F_G(U, t)$ denotes the t-affect value on the i-th vertex.
The proof is omitted here to save space.

We next utilize the t-affect function defined in Theorem 1 to design our relationship classification algorithm (RCA). RCA is composed by t rounds. In each round, we process $G(v_i \bigcup N(v_i))$ once and only once for each user $v_i \in V$. When we process $G(v_i \bigcup N(v_i))$, each user $u_j \in N(v_i)$ is assigned a unique *local index* $D_{v_i}(u_j)$ from $\{1, \cdots, |N(v_i)|\}$.

Recall that $G(N(v_i))$ tends to form disjoint clusters where the users having the same relationship with v more likely belong to the same cluster. Assume we are given a set of pre-labeled users V_i which have the same relationship ℓ_i with v_i, then we can utilize the t-affect function defined in Theorem 1 to explore the users who are t-affected by V_i most heavily. And the links between those users and v_i will be assigned the same relationship ℓ_i. However, in our problem setting, we are only given a set of pre-labeled *links* instead of users. Then we need to convert the pre-labeled information on links to that on vertices first. To this end, we define a vector A for each $G(N(v_i))$ and each labeling ℓ_i where

$$A[D_{v_i}(u_j)] = \mathcal{T}(\ell_i, v_i u_j)$$

Recall that initially $\mathcal{T}(\ell_i, e_j) = 1$ if e_j is pre-labeled by ℓ_i; 0, otherwise.

To ensure the consistency among the results gained from different users, whenever we have new information about any link's labeling, we should update the corresponding entry in SIT. In specific, for each user v_i, at each round, evaluating the t-affect range is interleaving with updating SIT which is illustrated as follows:

1) $A[D_{v_i}(u_j)] = \mathcal{T}(\ell_i, v_i u_j)$
2) $\mathcal{T}(\ell_i, v_i u_j) =$ the $D_{v_i}(u_j)$ th entry of

$$1_{N(v_i)} \times \left[I_{|N(v_i)|} - \prod_{j=0}^{1} \left(1_{N(v_i)} - A \cdot M^j_{G(N(v_i))}\right)_Z \right]$$

We get the final SIT after t rounds of computation. Each entry $\mathcal{T}(\ell_i, e_j)$ stores the confidence score for setting $\ell(e_j) = \ell_i$, it basically indicates the confidence level at which we label e_j using ℓ_i. Specially, if every link can only have one label, RCA will choose $\text{argmax}_{\ell_j \in \ell}\{\mathcal{T}(\ell_i, e_j)\}$ to label e_j at the highest confidence level.

One problem may arise here is how to determine the value of t. In other words, what is the minimum t such that ceratin accuracy level can be achieved with high probability. We will investigate this parameter setting in the performance evaluation part. Typically, when the network size is $60,000$ and 28% links are pre-labeled, it takes at least 15 rounds to achieve the best accuracy(around 80%).

Note that since our algorithm can be easily implemented in a parallel way, it works quite well in large scale online social networks. Please refer to Algorithm 1 for details.

IV. Marketing Strategies for Product Information Propagation

In most existing works on influence maximization problem, the relationship between two users is considered as a boolean one: people do not distinguish these relations. However, in reality, various relationships may play different roles in a particular information propagation. For example, a video game is more likely to be promoted among high school classmates

Algorithm 1 Relationship Classification Algorithm

Input: A online social network $G = (V, E)$ with partial pre-labeled links, and integer t for random walk.
Output: The online social network with all links labeled.

```
We consider each vertex $v_i \in V$, assign a unique local index $D_{v_i}(u_j) \in \{1, \cdots, |N(v_i)|\}$ to every neighbor $u_j \in N(v_i)$.
In the initial phase, we set
    $\mathcal{T}(\ell_i, e_j) = \begin{cases} 1 & \text{if } e_j \text{ is pre-labeled by } \ell_i \\ 0 & \text{otherwise.} \end{cases}$
repeat
    $t = t - 1$
    for each node $v_i$ do
        for each labeling $\ell_i$ do
            Define a vector $A$ where
            $A(D_{v_i}(u_j)) = \mathcal{T}(\ell_i, v_i u_j)$
            for each $u_j \in N(v_i)$ do
                $\mathcal{T}(\ell_i, v_i u_j)$ = the $D_{v_i}(u_j)$-th entry of
                $1_{N(v_i)} \times \left[ I_{|N(v_i)|} - \prod_{j=0}^{1} \left( 1_{N(v_i)} - A \cdot M^j_{G(N(v_i))} \right)_Z \right]$
                .
until $t = 0$;
for each link $e_j \in E$ do
    Label $e_j$ using $\text{argmax}_{\ell_j \in \ell}\{\mathcal{T}(\ell_i, e_j)\}$.
```

rather than among family members. Thus, a good propagation strategy should take different relationships into account. Specifically, depending on the particular information we want to propagate and the relationship of each link, we assign different probabilities to each link accordingly. We will discuss in details how to decide these probabilities in the experiment part. We first introduce the problem studied in this section:

Question 2 (product information propagation): In an online social network G, each node v is associated with two parameters: $w(v)$ denotes the weight of node v, reflecting v's importance, and $c(v)$ represents the cost by selecting v as one of the initial active users. Abusing notion, we use $c(S)$ to represent the total cost by selecting S as set of initial active users and $w(S)$ to represent the expected gain by selecting S as initial active users. Then under certain budget constraint $c(S) \leq B$, we try to select a set of initial active users $S \subseteq V$ such that the expected weight of final propagated nodes is maximized. For a budget constraint $B > 0$, the optimal solution of our weighted propagation maximization problem is:

$$OPT = \arg \max_{S \subseteq V : c(S) \leq B} w(S)$$

Obviously, the gain by selecting a set of active users depends on the social network structure and propagation model. In this work, we study the information propagation under *Independent Cascade Model* [1]: We start with a set of initial active nodes S_0, and the process unfolds in discrete steps according to the following randomized rule. When user v first becomes active in step t, it is given a single chance to activate each currently inactive neighbor w; it succeeds with a probability $p_{v,w}$, a parameter depending on the labeling of edge (v, w). If w has multiple newly activated neighbors, their attempts are sequenced in an arbitrary order. If v succeeds, then w will become active in step $t + 1$; but whether or not v succeeds, it cannot make any further attempts to activate w in subsequent rounds. Again, the process runs until no more activations are possible.

Given an initial active set S, Kempe *et al.* [1] propose an effective method to estimate the *expected influenced size* $w(S)$. Here we briefly introduce their main idea as follows. For each pair of neighbors (v, w) in the graph, a coin of bias $p(v, w)$ is flipped on edge (v, w). With all the coins flipped in advance, the process can be viewed as follows. The edges in G for which the coin flip indicated an activation will be successful are declared to be *live*; the remaining edges are declared to be *blocked*. Then a user u ends up active if and only if there is a path from some user in S to u consisting entirely of live edges. By repeating this process for sufficient number of rounds, we are able to estimate the expected weight $w(S)$ accurately.

Algorithm 2 Hill-Climbing Algorithm

```
$S_{[1]} := \arg\max_{v \in V; c(v) \leq B}\{w(\{v\})\}$;
$S_{[2]} := \emptyset$;
for $v \in V \setminus S_{[2]}$ do
    $v \leftarrow \arg\max_{v \in V; c(S_{[2]}) + c(v) \leq B}\{\frac{w(S_{[2]} \cup \{v\}) - w(S_{[2]})}{c(v)}\}$;
    $S_{[2]} = S_{[2]} \cup v$;
$S := \arg\max_{S \in \{S_{[1]}, S_{[2]}\}}\{w(S)\}$;
```

By assuming we are able to estimate the expected weight for any given initial active set, Algorithm 2 first computes two candidate sets for S: The first candidate set $S_{[1]}$ contains a single element which can maximize the total weight; the second candidate set $S_{[2]}$ is computed by Hill-Climbing Algorithm in which we always add the node v that can maximize the expected incremental marginal gain: $[w(S_{[2]} \cup \{v\}) - w(S_{[2]})]/c(v)$ until the budget constraint is violated. Then we choose the better one as the set S.

The expected weight of the propagated users resulted from S is at least $\frac{1}{2} \cdot (1 - \frac{1}{e})$ of the maximum possible weight under the same budget constraint. The proof is omitted here.

V. Experiment Results

Data Source: We extract the data from two real life online social networks to do evaluation. The first dataset is from the Renren net which contains 60,000 online users' profile. Each user profile contains the following features: User ID, Name, College School and the relationship to all his/her contacts. Renren is a Chinese social networking site with an interface similar to that of Facebook. It is popular among college students in China. It currently has more than 100 million registered users. Among all the links from Renren net, 96% belong to one of the four relationship classes: college classmate, highschool classmate, officemate and family

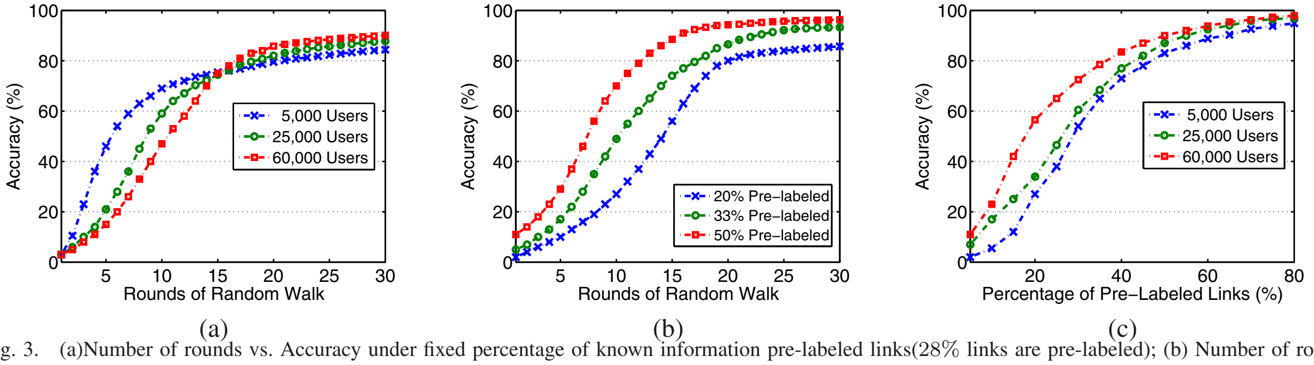


Fig. 3. (a)Number of rounds vs. Accuracy under fixed percentage of known information pre-labeled links(28% links are pre-labeled); (b) Number of rounds vs. Accuracy under fixed network size(60,000 users); (c)Percentage of pre-labeled links vs. Accuracy under fixed rounds t(t=5, 10, 15 for different network size).

member. In this experiment, we will only study these four relationships, the other labellings are ignored. We model it as a network where each node represents an online user, and the edge between a pair of users denotes that they have *contacts* with each other. The second set of data is from MSN users. We launch an online survey for 3 months, and collect the information from around 2000 online users successfully. The main goal of this survey is to estimate the propagation probability of specific product through particular relationship. These results are critical to evaluate the impact of relationship classification on product information propagation.

We summarize our experiment results for relationship classification and influence maximization in following subsections.

A. Relationship Classification

We conduct a series of experiments in order to evaluate the performance of RCA. In particular, we try to find out how the number of rounds t in Algorithm 1, the percentage of pre-labeled links and network size scale with the accuracy of our results.

Definition 1: Throughout the experiment results, we define **accuracy** as the ratio of the number of links which are labeled correctly by our method to the total number of unlabeled links.

The goal of the first experiment is to test the impact of the number of rounds t in Algorithm 1 on the accuracy under fixed percentage of pre-labeled links. We launch this experiment on three networks from Renren net with different sizes: 5000 users, 25,000 users and 60,000 users. In our experiment setting, we randomly pick 28% links as pre-labeled links, and the objective is to label the rest links with high accuracy. Figure 3(a) illustrates that for all those three networks, when the t exceeds certain threshold, the accuracy will achieve a stable level. In particular, for the network with 5000 users, it takes around 5 rounds to get accuracy level at least 80%, similarly, the network with $25,000$ users needs 10 rounds, and the largest network requires 15 rounds to get a satisfied accuracy. The main reason behind this result is that, the larger the network size is, the larger the diameter of the network will be. Since the percentage of pre-labeled links is fixed, if the diameter of the network is larger, it must take a longer time to propagate the pre-labeled information to farthest unlabeled links.

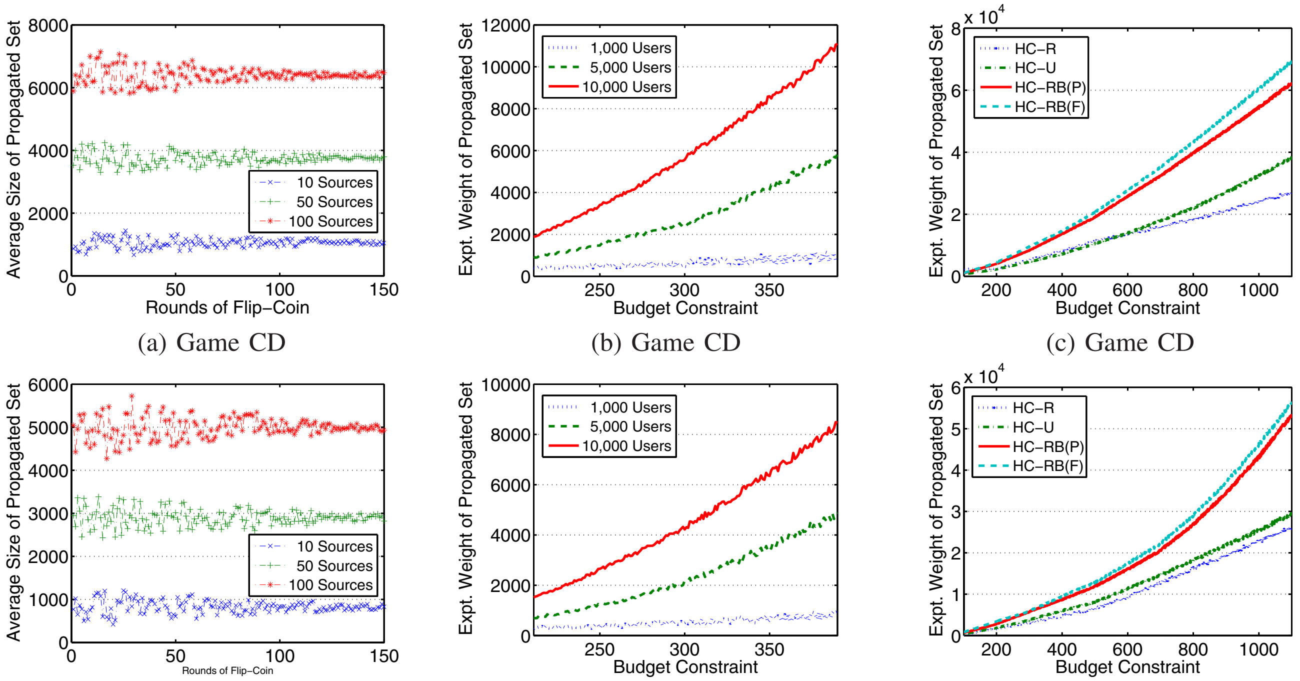
In the second experiment, we fix the network size and illustrate how the number of rounds t affects the accuracy under different percentage of pre-labeled links. By fixing the network size to $60,000$, we conduct three experiments under three different percentage of pre-labeled information. In particular, we randomly pick 20%, 33% and 50% links as pre-labeled links. And the curves in Figure 3(b) suggests that, when the known links is 20%, it takes 20 rounds to achieve accuracy level 70%; when the percentage of known links is 33%, we need 15 rounds to get accuracy level 85% ; with 50% known information, it takes 13 rounds to reach the accuracy level 90%.

In the third experiment, we fix the number of rounds t and try to find out how the accuracy will change under different percentage of pre-labeled links. We run this testing under three different network sizes 5000, 25,000 and 60,000. Based on the results from the first experiment, in order to find the minimum number of rounds t which can achieve the stable accuracy, we adjust t according to different network size, *e.g.*, $t = 5, 10, 15$ correspondingly. The results from Figure 3(c) show that, for all the three networks, the accuracy level is able to achieve 80% as long as percentage of the pre-labeled links is at least 30%. The important insight here is if the number of rounds t is chosen appropriately according to different network size, then the accuracy level does not depend on the network size but only the percentage of pre-labeled links. In other words, our algorithm is robust to the network size.

B. Relationship Based Propagation

In this experiment, we try to evaluate the impact of relationship classification on the product information propagation process. Based on our survey from more than 2000 MSN users, we get a rough idea of the propagation probability through different relationship regarding to several different products, the main results are listed in Table. 2.

The online survey is conducted as follows: For each attended user, he/she is given a list of products and a list of relationships, and the system will let him/her select whether he/she will recommend particular product to his/her contact with particular relationship if they meet online. By summarizing all the selections, for each product-relationship pair, we



(a) Game CD

(b) Game CD

(c) Game CD

(d) Printer

(e) Printer

(f) Printer

Fig. 4. (a)(Game CD)Number of flip-coin rounds Vs. Computed size of propagation set under various size of initial active set; (b)(Game CD)Budget Constraint Vs. Weight of Propagation Set under various network size; (c)(Game CD)Budget Constraint Vs. Weight of Propagation Set under various schemes. HC-RB(F) denotes the relationship-based scheme with full knowledge; HC-RB(P) denotes the relationship-based scheme with 30% pre-labeled links; HC(U) denotes the uniform probability assignment; HC(R) denotes the randomized probability setting; (d)(Printer) Same information as (a); (e)(Printer) Same information as (b); (f)(Printer) Same information as (c).

use the ratio of the number of users choosing yes to the total number of users as the propagation probability.

	Game CD	Cleaner	T-Shirt	Book	Printer
Highschool	10%	2%	15%	30%	20%
College	55%	5%	30%	40%	20%
Officemate	45%	25%	30%	30%	45%
Family	5%	45%	40%	30%	20%

TABLE II
FOR EACH PRODUCTION-RELATIONSHIP PAIR, THE CORRESPONDING ENTRY DENOTES THE PROPAGATION PROBABILITY.

Experiment setup: We still use the data crawled from Renren net as the underlying social network to evaluate our algorithm on product propagation problem. Depending on the specific product we desire to promote, we carefully assign the propagation probability to each relationship based on our survey results. Then we are able to get a *propagation network* with respect to that particular merchandize. For simplicity of analysis, each user is randomly assigned a weight from $\{1, \cdots, 200\}$ and a cost from $\{1, \cdots, 100\}$.

Experiment results: In the following experiments, we choose two products as an example to evaluate the performance of our algorithm and illustrate the impact of relationship classification on the propagation performance.

In the first set of experiments, we desire to promote Game CD to the market under certain budget constraint. Based on the data from Renren net and our online survey, we can immediately construct a *propagation network* with respect to Game CD. In the propagation network, we assign probability 55%, 10%, 45% and 5% to each link connecting two college classmates, highschool classmates, officemates and family members accordingly. We first assume that *all* the links are pre-labeled, that is the propagation probability on each link is pre-known. Then we implement Algorithm 2 to find the initial active set under various budget constraints.

Remember that in order to evaluate the expected size of propagation set for any given set of initial active users, we will flip coin with bias equal to its propagation probability on each link. Then after sufficient number of rounds, the average number of users that can be reached by the initial active users(through live edges) is very close to the expected propagation size. The problem here is what is a appropriate number of rounds to flip coin on each link? How does it scale with the size of initial active set? For simplify of analysis, we will not consider the weight and cost associated with each node at this step. By fixing the network size to 10,000 users, we conduct this experiment under three different sizes of initial active set: 10, 50, 100. The results in Figure 4(a) give us a brief idea of this problem: Typically, the more initial active nodes we have, the more number of rounds is required to achieve certain accuracy. For the initial active set with 100 users and 50 users, we need at least 100 rounds to get a accurate estimation. In contrast, we need only 50 rounds to get a satisfied estimation under initial active set with 10 users. This result serves as a guideline to choose an appropriate number of flip-coin rounds in our following experiments. Through the following experiments, we set the number of flip-coin rounds to 80 in order to balance the computation complexity and estimation accuracy. This parameter can be adjusted dynamically according to various requirements.

In the second experiment, we implement Algorithm 2 to find initial active set under various budget constraints. Notice that in this experiment, we assume all links are pre-labeled and the propagation probability in each link is pre-known. Figure 4(b) demonstrates how the propagation gain scales with the budget constraint. We conduct this experiment under three network sizes: 1,000 users, 5,000 users and 10,000 users.

The objective of the third experiment is to evaluate the impact of relationship classification on production propagation problem. Note that in previous experiment, we assume all the links are pre-labeled and propagation network is pre-known completely. In this experiment, we assume only 30% of the links are pre-labeled. Our two-phase relationship-based scheme works as follows: 1) we first conduct RCA to predict the relationship on each unlabeled link and the propagation network can therefore be constructed; 2) we next select the initial active set by running Algorithm 2 on previous predicted propagation network.

We compare our scheme with three other schemes: 1)Relationship-based scheme with full knowledge: assume all links are pre-labeled and propagation network is pre-known, Algorithm 2 is implemented to find the initial active set; 2)Uniform probability assignment: the propagation probability of each link is identical(it is set to 50% in our experiment), and we also run Algorithm 2 to find the initial active set. This scheme does not take the impact of different relationships into account; 3)Randomized probability assignment: the propagation probability of each link is randomly chosen from 1% to 100%. Again, this probability assignment does not consider the impact of relationship classification either.

We then evaluate the expected propagation gain resulted from those four initial active sets. Notice that all the evaluations are conducted on the *real* propagation network (with all links pre-labeled). The results in Figure 4(c) confirm that both relationship-based schemes beat the other two schemes. And relationship classification indeed plays a critical role in product information propagation. For those two relationship-based schemes, the more the pre-labeled links we have, the better the performance will be. Surprisingly, when only 30% links are pre-labeled, the propagation gain of our two-phase scheme is very close to the one with full knowledge. In the second set of experiments, we basically conduct same series of testing on promoting Printer. Please refer to Figure 4(d)(e)(f) for detailed results and we gain similar insights from those results.

VI. Related Work

Holme *et al.* analyze an online dating community in detail [2]. Mislove *et al.* investigate not only online SNS, but also other web services that have social networking features [3]. Java *et al.* conducted a research on microblogs [4]. In their work, the in-degree and out-degree distribution of the network, and the activity pattern of users were analyzed. Leskovec *et al.* analyzed the largest social network (instant messaging network) [5]. They analyzed various aspects such as the number of buddies, the duration of conversations. Various models have been developed for relational learning. A notable study is that of Probabilistic Relational Models (PRMs) [6]. Such models provide a language for describing statistical models over relational schema in a database. Perlich *et al.* [7] also propose aggregation methods in relational data. Backstrom *et al.* [8] analyzes community evolution, and shows that some structural features characterizing individuals' positions in the network are influential, as well as some group features such as the level of activity among members. Cai *et al.* [9] studied the community mining from multi-relational networks. For the influence maximization problem, Domingos and Richardson [10] [11] initially study it as an algorithmic problem. Kempe, Kleinberg, and Tardos [1] first formulate the problem as a discrete optimization problem and they give a simple greedy method which can achieve constant approximation. Several studies improve the efficiency of greedy algorithm ([12]).

VII. Conclusion

In this paper, we proposed efficient and effective methods for link classification in online social networks and for maximizing the influence in such networks. We would like to continue to improve the efficiency and efficacy of our methods for link classification and influence maximization problems.

VIII. Acknowledgement

The research of authors is partially supported by NSF CNS-0832120, NSF CNS-1035894, National Basic Research Program of China (973 Program) under Grant No.2010CB334707, National Natural Science Foundation of China under Grant No. 60828003, No.60773042, the Natural Science Foundation of Zhejiang Province under Grant No.Z1080979, program for Zhejiang Provincial Key Innovative Research Team, program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program).

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD' 03*.

[2] P. Holme, C. Edling, and F. Liljeros, "Structure and time evolution of an Internet dating community," *Social Networks*.

[3] A. e. Mislove, "Measurement and analysis of online social networks," in *IMC '07*.

[4] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *SNA-KDD '07*.

[5] J. Leskovec and E. Horvitz, "Planetary-scale views on a large instant-messaging network."

[6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI 1999*.

[7] C. Perlich and F. Provost, "Aggregation-based feature invention and relational concept classes," in *KDD '03*.

[8] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *KDD '06*.

[9] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Community mining from multi-relational networks," *LNCS*.

[10] P. Domingos and M. Richardson, "Mining the network value of customers," in *KDD '01*.

[11] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *KDD' 02*.

[12] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *KDD*. ACM, 2009, pp. 199–208.