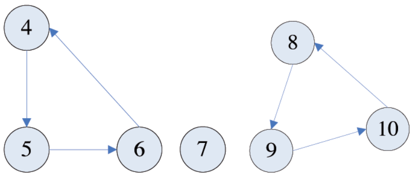
Approaches of influence maximization in social networks with positive and negative opinions



ENFOQUES PARA MAXIMIZAR LA INFLUENCIA EN LAS REDES SOCIALES CON OPINIONES POSITIVAS Y NEGATIVAS

DOI: http://dx.doi.org/10.6036/7583 | Recibido: 10/mar/2015 • Aceptado: 25/may/2015

Jiaguo Lv[1,2,*], Jingfeng Guo[2,3], Yuanying Liu[2], Wei Zhang[1] and Allen Jocshi[4]

[1] School of Information Science and Engineering, Zaozhuang University, Zaozhuang, Shandong, 277100, China
[2] School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, 066000, China
[3] The Key Laboratory for Computer Virtual Technology and System Integration Hebei Province, Qinhuangdao, Hebei, 066004, China
[4] Network Information Center for Design and Analysis, MCCN Ltd,11952,Gdansk, Poland
* Corrpesonding Author, lvjiaguo2004@163.com

RESUMEN

- En este trabajo se propone un nuevo modelo de correlación lineal para opiniones negativas (Lineal Threshold with Negative opinions - LTN) basado en el modelo básico de correlación lineal (LT), para considerar el fenómeno de las opiniones negativas que pueden aparecer y propagarse en las redes sociales como un fenómeno viral. Complementariamente se muestran algunas propiedades del modelo LNT como la monotonicidad y la submodularidad. Con estas propiedades se propone un algoritmo cercano al Greedy con una relación de 1-1e para maximizar la influencia en el modelo LNT. Para superar la ineficiencia del algoritmo Greedy, se han utilizado en este trabajo tres algoritmos mejorados: el algoritmo Nuevo Greedy para LNT, el algoritmo CELF para LNT y el algoritmo Mixed Greedy para LNT. Los resultados experimentales con dos series de datos mostraron que la extensión de influencia calculada con estos algoritmos era similar a la de los algoritmos comparativos pero siendo mucho más rápidos que estos algoritmos comparativos.
- **Keywords:** marketing viral, maximización de la influencia, redes sociales, opiniones negativas, modelo LTN.

ABSTRACT

In viral marketing, considering the phenomenon that negative opinions may emerge and propagate in social networks, based on the fundamental linear threshold model (LT), a new model – linear threshold model with negative opinions (LTN) was proposed in this study. Subsequently, some properties of the LTN model, such as monotonicity and submodularity have been shown. With these properties, a greedy approximate algorithm with a ratio of (1-1/e) for influence maximization on the LTN model was proposed. To overcome the inefficiency of the greedy algorithm, three improved algorithms—LTN_NewGreedy (NewGreedy algorithm on LTN), LTN_CELF(CELF algorithm on LTN) and LTN_MixedGreedy (MixedGreedy algorithm on LTN) have been provided in this work. The experimental results on two synthetic datasets showed that the influence spread of these improved algorithms was close to that of those benchmark algorithms, but they were faster than those benchmark algorithms.

Keywords: Viral marketing, Influence maximization, Social network, Negative opinions, LTN model.

1. INTRODUCTION

Based on social influence among an individual's circles of friends, families and so on, viral marketing is believed to be an effective marketing strategy. With the increasing popularity of large-scale social networking sites, such as Facebook, Twitter and so on, viral marketing has more potential than ever before. Two key problems that would enable such large-scale online viral marketing are modeling influence propagation and influence maximization. The literatures regarding these two problems is extensive, but most of the previous works ignore an important fact that we often experience in the real word. That is, not only will the positive opinions propagate in the network, but also the negative opinions. Taking the phenomenon into consideration, Chen et al. [3] and Nazemiam and Taghiyareh [20], proposed the extended influence propagation models based on the fundamental independent cascade model (IC), and proposed algorithms for influence maximization based on their models. In this paper, an extension of the LT model that incorporates the propagation of negative opinions named LTN was proposed. The new model maintains some nice properties, such as monotonicity and submodularity, which allows an approximate greedy algorithm for influence maximization with a ratio of (1-1/e) . To improve the efficiency of the greedy algorithm, some improved algorithms have been proposed too.

The remainder of the paper is organized as follows: section 2 surveys related work; section 3 presents the LTN model and its properties; section 4 details greedy algorithm and our improved algorithms for influence maximization based on the LTN model. We report a performance evaluation in section 5. Finally, we offer conclusions in section 6.

2. RELATED WORK

In this section, we first detailed the related work of the diffusion model of influence and influence maximization, then we listed the notations used in this paper.

2.1 DIFFUSION MODEL OF INFLUENCE

In [25], a SEIR model was proposed to describe epidemic of the virus on the online social network. Hower, the most fundamental influence models in the literature that capture the underlying dynamics of the diffusion process are the LT model and the IC model. The LT model was proposed by Granovetter et al. [1] and generalized by Watts et al. [2]. In this model, at any time step, a node is either active or inactive. Once a node is activated in a time step, it will remain active forever in the whole propagation. In the LT model, the sum of incoming edge weights on any node is assumed to be at most 1 and every node chooses an active threshold uniformly at random from [0,1]. At any time step, if the sum of incoming influence (edge weights) from the active neighbors of an inactive node exceeds its threshold, it becomes active.

However, the LT model ignores the propagation of the negative opinions in the network. Incorporating the negative opinions, based on the IC model, Chen et al. [3] proposed the IC-N model and MIA-N algorithm for influence maximization in IC-N. Similar to Chen et al. [3], Nazemian et al. [20] proposed an extended model ICPN with positive and negative WOM (word of mouth) based on the IC model, and proposed a greedy algorithm with approximation ratio of (1-1/e). Similar to the IC-N model, in this paper, based on the LT model, we propose the LTN model in this work, and the model will be detailed in section 3.

2.2 INFLUENCE MAXIMIZATION

Influence maximization is the problem of finding a small subset of nodes (seed nodes) in a social network that could maximize the spread of influence. Domingos and Richardson [4] and Richardson and Domingos were the first to study it as an algorithmic problem. Their methods are probabilistic. Kempe et al. [6] were the first to formulate it as a discrete optimization problem. The problem under the IC model and LT model was proved to be NP-hard, and a general greedy algorithm (KK_Greedy) with an approximation ratio of 1-1/e was proposed [6]. Due to the inefficiency of the KK_Greedy, considerable work has been done to improve its efficiency [7-17]. Lu et al. [18] studied the complexity of the influ-

Terms	Descesription
S	Seed node set (in algorithm1, algorithm2, algorithm3, algorithm4)
K	The size of seed node set (in algorithm1, algorithm2, algorithm3, algorithm4)
getPosInfluenceSet(G,S,q)	A function which returns the positive influenced node set of S in G under the quality factor q (in algorithm1, algorithm 3, algorithm 4)
σ_G(S,q)	The size of the positive influenced set of the seed set S in G under the quality factor q (in section 3.1)
F_{Gi}(S)	The reachable set from S in G_i (in section 4.2.1)
MG(G,S,v)	MG(G,S,v)=|getInfluenceSet(G,S+{v})- getInfluenceSet (G,S)| (in algorithm2, algorithm4)
$G_i=(V_i,E_i)$	The random graph that we get by running a random live edge selection process on G in iteration i (in algorithm2, algorithm4)
$G_i^S(V_i^S, E_i^S)$	The induced graph from G_i ,where V_i^S =V_i\ F_{Gi}(S), E_i^S ={(u,v)|u,v∈ V_i^S ,(u,v) ∈ E_i } (in algorithm2, algorithm4)
SCC_i	The macro node that denotes the ith strong connected component in the induced graph G_i^S (in algorithm2, algorithm4)
sccCount	The number of strong connected components in graph (in algorithm2, algorithm4)
u.mg1	The property of node u to denote the marginal gain of u for the current iteration (in algorithm2, algorithm4)
u.mg	The property of node u to denote the expected marginal gain of u for all iterations (in algorithm2, algorithm4)
Q<u,u.mg,u.mgset,u.flag>	A table for all candidate nodes. In Q, u.mgset is the marginal influenced set of node u for the current s, that is, u.mgset = getInfluenceSet(g,S+{u})-getInfluenceSet(g,S), u.mg= |u.mgset|, and u.flag is the number of iteration when u.mg was last updated. (in algorithm3, algorithm4)

Table 1 Notations

ence maximization problem in deterministic linear threshold model, and showed that the exact computation of the exact computation of the influence given a seed set can be solved in polynomial time.Bharathi et al. have studied the competitive influence diffusion with an extension of the IC model [19]. The algorithmic perspective of the negative opinions' diffusion has been discussed [3,20].Their work is based on an extended version of the IC model, while our work is based on an extended model of the LT model.

2.3 NOTATIONS

For ease of reading, we list the notations used in this paper in Table 1.

3.THE LTN MODEL AND ITS PROPERTIES

In this section, we first introduce the LT model and IC model, then propose the LTN model. Finally, we will provide some useful properties of LTN.

3.1 LT MODEL

In our work, social network is modeled as a directed graph G= (V, E), where V and E are the node set and edge set, respectively. Every u $\in$ V denotes an individual, and a directed edge (u,v) $\in$E represents the influence from u to v, and the volume of the influence is denoted by the edge weight of (u,v).

LT model(Linear threshold model) is a simple version of threshold model. In the LTN model, every node has two possible states, namely,active and inactive. In some situations, when node v adopts a new product or adopts a new technology, the node is active. On the other hand, it is inactive. In LT model, once the node is active, its state will never change. So, the LT model is progressive.In this model, each node v$\in$V has a nonnegative weight w_{uv}. For every u$\in$$N^{in}$(v), $\sum_{u\in N^{in}(v)} W_{uv} \le 1$.For every node v in V, it has a personal threshold value θ. Given these thresholds and an initial set S of active nodes (seed set), the diffusion process unfolds deterministically in a sequence of steps. At the time step t, each node which was active at time t-1 automatically remains active. Each node v that was inactive at time t-1 becomes active at time t if and only if $\sum_{u\in N^{in}(v),u\in S} W_{uv} \ge v.q$. Intuitively then, the edge weight represents the extent to which v is influenced by u, and the threshold represents the personal tendency of v to adopt a new technology when its in-neighbor do.

Let S is the given seed set, $S_{inactive}$ is the set of nodes whose state is inactive and S_{new} is the set of new active nodes in the current time step. The propagation of the LT model is described below.

```
$S_{inactive}$=V-S; $S_{new}$=S;
While |$S_{new}$|>0 {
  $S_{new}$={}
For each node v in $S_{inactive}$ do {
    If $\sum_{u\in S,u\in N^{in}(v)} W_{uv} \ge v.\theta${

      S= S+{v}
$S_{new}$= $S_{new}$+{v}
$S_{inactive}$= $S_{inactive}$-{v}
 }
}
```

3.2 IC MODEL

Inspired by the research on interacting particle systems, the indepedent cascade model (IC) of diffusion was studied widely in the field of influence maximization. Same to the LT model, in the IC model, a node has two possible states, active and inactive. In the IC model, each new active node has a single chance to activate each of its inactive out-neighbor node. Moreover, the probability that a node is activated by a new active in-neighobr is indepedent of the set of neighbors who have attempted to active it in the past. Once again, starting with an initial seed set S, the propagation unfolds in a series of steps. At step t, any node u who has just become active tries to activate each of its inactive out-neighbors v. Then, at the time step t+1, v will be active with probability $p_{u,v}$. Whether or not v becomes active, u will never activate v throught the rest of the diffusion process.

3.3 LTN MODEL

In real world, when we receive some products and services, both positive opinions and negative opinions may emerge and propagate. However, in the LT model, only positive opinions have been taken into account. Therefore, when studying the influence maximization problem, it shoule be important to incorporate the conatgion of negative opinions into the LT model. So, taking into account of the negative opinions, a new threshold model LTN is proposed. In the LTN model, when the node is inactive, it is in neutral state. When a neutral node is influenced by positive opinions, it will be in positive state. On the contrary, if a node is influenced by negative opinions, the node will be in negative state. Both negative and positive are active states. Same to the LT model, the LTN model is progressive too. That is , if a node is in positive or negative state, it will never be in neutral state. Similar to the LT model, in the LTN model, every node has threshold. However, in LTN model, a neutral node may be influenced by nagative opinions or positive opinions. So, every node in the LTN model has two thresholds, θ_N and θ_p. The former is the threshold for its negative in-neighbors, and the latter is the threshold for its positive in-neighbors. As shown by Rozin and Royzman [21], there is negative bias in social psychology. To match this phenomenon, in the LTN model, θ_N is not greater than θ_p (θ_N ,$\theta_p$$\in$[0,1]). A discrete time step is used to model the dynamic change in the network. The model has a parameter q (q$\in$[0,1]) called quality factor, which is the probability that a neutral node switches to positive when it is activated by its positive in-neighbor nodes. For an initial seed set S, at the beginning, t= 0, all nodes in S are positive and all nodes in V\S are neutral. At time step t, for a neutral node v, $PA_t(v) \subseteq N^{in}(v)$ is the positive in-neighbor set of v, and $NA_t(v) \subseteq N^{in}(v)$ denotes the negative in-neighbor set of v. If $\sum_{u\in NA_t(v)} W_{uv} \ge v.\theta_N$, v will be negative, otherwise, if $\sum_{u\in PA_t(v)} W_{uv} \ge v.\theta_P$, v will be activated by its positive in-neighbous, and it will be positive with probability q, and will be negative with probability 1-q. When there are no new active nodes in a time step, the activation process will stop.

Due to the incorporation of the propagation of negative opinions,in essence, the LTN model is similar to the IC-N model proposed in [3]. However, the IC-N model is an extension of the IC model, the LTN model is an extension of the LT model.

Let S is the given seed set, S_N is the set of nodes whose state is negative, S_P is the set of nodes whose state is positive, $S_{inactive}$ is the set of nodes whose state is neutral and S_{new} is the set of new active nodes in the current time step. The propagation of LTN model is described below.

```
$S_N$={}; $S_P$=S; $S_{inactive}$=V-S; $S_{new}$=S;
While | $S_{new}$ |>0 {
$S_{new}$={}
For each node v in $S_{inactive}$ do {
    If $\sum_{u\in NA_t(v)} W_{uv} \geq v.\theta_N$ {
        $S_N$= $S_N$+{v}
        $S_{new}$= $S_{new}$+{v}
$S_{inactive}$= $S_{inactive}$-{v} }
    Else if $\sum_{u\in PA_t(v)} W_{uv} \geq v.\theta_P$ {
        $S_P$= $S_P$+{v} with probability q
$S_N$= $S_N$+{v} with probability 1-q
$S_{new}$= $S_{new}$+{v}
$S_{inactive}$= $S_{inactive}$-{v} }//if
}//For
}//While
```

Obviously, when q is 1, the LTN model is degraded into the LT model. The positive influence spread of a seed set S in G with quality factor q is the expected number of positive nodes in the graph when the activation process stops, and is denoted as $\sigma_G(S,q)$. In this work, we take positive influence spread as our objective since it is directly related to the expected revenue that the marketer would gain from the viral marketing. The influence maximization under the LTN model is described as follows.

For a given network G, size of seed set K and quality factor q, the influence maximization under the LTN model is to find a seed set S* of cardinality k, such that for any node set S of cardinality k, $\sigma_G(S^*,q) \geq \sigma_G(S,q)$ holds, in other words, $S^*= \arg\max_{S\subseteq V,|S|=k} s_G(S,q)$.

3.4 PROPERTIES OF THE LTN MODEL

Now, we discuss several important properties of $\sigma_G(S,q)$, which will be used in section 4. Firstly, we present a random selection process of live edges.

Definition 1. Selection of live edges. Recall that each node v has an influence weight $W_{uv} \geq 0$ from each of its in-neighbors u, subject to $\sum_{u\in N^{in}(v)} W_{uv} \leq 1$. Suppose that v picks at most one of its incoming edges at random, selecting the edge from u with probability $W_{uv}*q$, and selecting no edge with probability $1-q*\sum_{u\in N^{in}(v)} W_{uv}$. The selected edge is declared to be 'live', and all other edges are 'blocked'.

Theorem 1. For a given seed set S, the following two distributions over sets of nodes are the same:

(1) The distribution over positive sets obtained by running the LTN process starting from S;

(2) The distribution over sets reachable from S via live-edge paths, under the random selection of live edge defined above.

Proof. We need to prove that reachability under our random choice of live edges defines a process equivalent to that of the LTN model.

First, we consider the diffusion process of the LTN model. We define S_t to be the set of positive nodes at the end of iteration t, for t = 0,1,2,…(S_0 = S). If node v has not been activated by the end of iteration t, then the probability that it becomes active in iteration t+1 is equal to the chance that the influence weights in $S_t \backslash S_{t-1}$ push it over its threshold, given that its threshold was not exceeded already; this probability is $\frac{q*\sum_{u\in S_t\backslash S_{t-1}} W_{uv}}{1-q*\sum_{u\in S_{t-1}} W_{uv}}$.

Second, we consider the reachability of the random selection of live edge. We run the live edge process by revealing the identities of the live edges gradually as follows. We start with the seed set S_0. For each node v with at least one edge from the set S_0, we determine whether v's live edge comes from S_0. If so, then v is reachable; otherwise, we keep the source of v's live edge unknown, subject to the condition that it comes from outside S_0. Having now exposed a new set of reachable nodes S_1' in the first stage, we proceed to identify further reachable nodes by performing the same process on edges from S_2', S_3',…. If node v has not been determined to be reachable by the end of stage t, then the probability that it is determined to be reachable in stage t+1 is equal to the chance that its live edge comes from S_t'\$S_{t-1}$', given that its live edge has not come from the earlier sets. So, the probability is $\frac{q*\sum_{u\in S_t\backslash S_{t-1}} W_{uv}}{1-q*\sum_{u\in S_{t-1}} W_{uv}}$, which is same as in the LTN model.

Thus, by induction over these stages, we see that the live edge process produces the same distribution over positive sets as the LTN model.

Theorem 2. For any arbitrary instance of the LTN model, if the quality parameter q is fixed, for the seed set S, the positive influence spread $\sigma_G(S,q)$ is monotone, submodular and $\sigma_G(\phi,q)=0$.

A set function on vertices of G=(V,E) is a function f: $2^V \rightarrow R$. Set function f is monotone, if $f(S) \leq f(T)$ for all $S \subseteq T$, and it is submodular if $f(S\cup\{u\})-f(S) \geq f(T\cup\{u\})-f(T)$ for $S \subseteq T$ and $u\in V\backslash T$.

Proof. (1) Monotone. When S is ϕ, there are no positive nodes at the beginning of the activation process, no neutral nodes will be influenced by positive nodes and new positive nodes will never come into being. So, $\sigma_G(\phi,q)=0$. Obviously, when q is fixed, $\sigma_G(S,q)$ is monotone. Because, when the new node $v \notin S$ becomes positive, it will influence its neutral out-neighbour nodes, and the probability of these nodes becoming positive will be increased. That is, $\sigma_G(S+\{v\},q) \geq \sigma_G(S,q)$.

(2)Submodular. To establish this result, we should consider the expression $\sigma_G(S\cup\{u\},q)-\sigma_G(S,q)$. In other words, what increase do we get in the expected number of overall positive nodes when we add v to the seed set S. It is difficult to analyze directly. Our proof deals with the difficulty by considering the equivalent live edge selection process with the LTN model. For a fixed outcome X of live edge selection, R(v,X) denotes the nodes that can be reached from v on a path consisting entirely of live edges.

First, we give the proof of the submodularity of $\sigma_X(S,q)$. Suppose, $S \subseteq T \subseteq V$, $\sigma_X(S\cup\{v\},q)-\sigma_X(S,q)= |R(v,X)\backslash \bigcup_{u\in S} R(u,X)| \geq |R(v,X)\backslash \bigcup_{u\in T} R(u,X)| = \sigma_X(T\cup\{u\},q)-\sigma_X(T,q)$, so$\sigma_X(S,q)$ is submodular.

Then, we consider the positive influence spread$\sigma_G(S,q)$. We have,

$$\sigma_G(S,q) = \sum_X \Pr ob(X) * \sigma_X(\mathrm{S,q}) \qquad (1)$$

In (1), prob(X) is the probability of X in its probability space. From (1) ,we know that $\sigma_G(S,q)$ is a non-negative linear combination of $\sigma_X(S,q)$. $\sigma_X(S,q)$ is submodular, so, $\sigma_G(S,q)$ is submodular too.

Theorem 3. The positive influence maximization problem is NP-hard for the LTN model.

Proof. For the influence maximization of the LT model, the proof of its difficulty is given in Kempe et al. [6]. Since the LT model is a special case of the LTN model when q is 1, the positive influence maximization for the LTN model is NP-hard too.

4. ALGORITHMS FOR INFLUENCE MAXIMIZATION UNDER THE LTN MODEL

With theorem 2 and the work of Kempe et al. [6], for the influence maximization under the LTN model, we can get a simple approximate KK-greedy algorithm with a ratio1-1/e.

4.1 KK-GREEDY ALGORITHM

KK-greedy algorithm is described in algorithm 1. The algorithm builds the initial seed set one node at a time, always greedily choosing the node with the largest marginal gain in influence.

Algorithm 1 kk_greedy

Input: Social network G, Size of seed set K, Quality factor q

Output: Seed set S

```
S=ϕ;
While |S|<K do {
    u=$\underset{v\in V\setminus S}{\text{argmax}}(|\,\text{getPosInfluenceSet}(G,\text{S}+\{\text{v}\},\text{q})-\text{getPosInfluenceSet}(G,\text{S},\text{q})\,|)$
    S=S∪{u}
}
```

In KK-Greedy, getPosInfluenceSet(G,S,q) is used to get the positive node set by S in G. The most major limitation of KK-Greedy is its inefficiency. The inefficiency is two-fold:

(1) The computation of getPosInfluenceSet(G,S,q) is computationally expensive with Monte Carlo simulation;

(2) There are too many candidate nodes needing to be examined by computing their marginal gain of influence.

4.2 IMPROVED ALGORITHMS FOR KK-GREEDY ALGORITHM

In recent years, considerable work has been done to improve the efficiency of KK-Greedy algorithm. To address the first issue, a lot of excellent algorithms have been proposed, such as NewGreedy [8] , and so on. To tackle the second issue, CELF [11] and CELF++ [12] have been proposed.

4.2.1 Improving the efficiency of influence function

As we know, the evaluation of the influence spread by Monte Carlo simulation is very inefficient. Based on theorem 1, we propose a new method to evaluate the influence spread, which is similar to the method of NewGreedy [8]. Similar to Monte Carlo simulation, we select an integer R as simulation times. In iteration i, we run the random live edge selection process on G, and get a graph G_i. Suppose, $F_{Gi}(S)$ is the reachable set from S in G_i, then, we have,

$$\sigma_G(S,q) \;=(1/R)*\sum_{i=1}^{R}|F_{G_i}(S)| \tag{2}$$

Like KK-Greedy, for the current seed set S, in each iteration, from all nodes in V\S, we greedily choose the node v with the maximal value of |getPosInfluenceSet(G,S+{v},q)-getPosInfluenceSet (G,S,q)|, and add it to S. For simplicity, MG(G,S,v,q) is used to denote the expression | getPosInfluenceSet (G,S+{v},q)- getPosInfluenceSet (G,S,q)|, F_G (S) is the reachable set from node set S in graph G.

For every graph G_i=(V_i,E_i) obtained from the random live edge process, with the concept strong connected component in graph theory, we may reduce the computational complexity of MG(G_i,S,v,q). With the concept of reachability and strong connected component, we know that, for a strong connected component SCC_i in graph G and any two nodes u and v (u,v∈ SCC_i and u≠v), F_G({u})= F_G({v}). The main idea of computing MG(G_i,S,v,q) is as follows.

(1) First, for the current seed set S ,we compute the reachable set F_{Gi}(S). Then, for any node v in S∪F_{Gi}(S), MG(G_i, S,v)=0.

(2) Suppose, V_i^S =V_i\ F_{Gi}(S), E_i^S={(u,v)|u,v∈ V_i^S ,(u,v) ∈ E_i }, then we can get the induced graph $G_i^S(V_i^S, E_i^S)$ from G_i.

(3) Get all strong connected components from G_i^S. From graph theory, we know that all nodes in a strong connected component have the same reachable set, therefore, we use a macro node to denote a strong connected component. Now, macro node SCC_i can be used to denote the strong connected component SCC_i. If there is an edge in G_i^S from nodes in strong connected component SCC_i to SCC_j, we will add an edge from macro node SCC_i to SCC_j, thus, from induced graph G_i^S,we can get a macro graph SCC_i^S .

(4) Compute the reachable set for every node in G_i^S. Let, $F_{SCC_i^S}(SCC_i)$ be the reachable set of macro node SCC_i in the macro graph SCC_i^S. Then for any node v in V_i^S, we have,

$$MG(G_i,S,v)=\sum_{scc\in \text{F}_{\text{SCC}_i^S}(SCC_i)}|scc| \tag{3}$$

Discussion. From the empirical work of social network, we know that there are a lot of certain-scale strong connected components in the graph. But, if this is not the case, the above method will be inefficient, and we will compute all reachable set for every node directly as NewGreedy [8]. So, in our algorithm, we introduce a threshold θ, when the number of strong connected component sccCount is less than θ*|V_i^S|, we will compute MG(G_i,S,v) with (3), otherwise, we will compute MG(G_i,S,v) directly. So, we have,

$$MG(G_i,S,v)=\begin{cases}\begin{cases}0 & \text{if v in S } \cup\text{F}_{\text{G}_\text{i}}\text{ (S) and sccCount} < \theta\text{*|V}_\text{i}^\text{S}\text{ |}\\ \sum\limits_{\text{scc}\in\text{F}_{\text{SCC}_\text{i}^\text{S}}\text{(SCC}_\text{i}\text{)}}|scc| & \text{if v in G}_\text{i}^\text{S}\text{ and sccCount} < \theta\text{*|V}_\text{i}^\text{S}\text{ |}\end{cases}\\ |F_{G_i^S}(\{v\})| \quad \text{if sccCount} \geq \theta\text{*|V}_\text{i}^\text{S}\text{|}\end{cases} \tag{4}$$

In order to obtain all strong connected components from the graph, we adopt Tarjan's algorithm [22], and its complexity is O(m+n). Our algorithm LTN_NewGreedy is described in algorithm 2.

Algorithm2 LTN_NewGreedy

Input: Social network G, Size of seed set K, Quality factor q, threshold θ

Output: Seed set S

```
S=ϕ;
For v in V do {v.mg=0}
While |S|<K do
```

```
{
  4: For i=1 to R do
   {
  5:Based on the process of live edge selection, we get the random graph $G_i$
  6:For the current seed set S, we compute the reachable set $F_{Gi}(S)$
       7:Get the induced graph $G_i^s=(V_i^s, E_i^s)$
       8:Get all strong connected components to SCCList
       9:Get the macro graph $SCC_i^s$
       10:For every node v in Vi, we compute MG($G_i$,S,v) with formula (4), v.mg1= MG($G_i$,S,v)
   }//For
  11:Get the average marginal gain of node v for R iterations, v.mg=(1/R)* $\sum_{i=1}^{R} v.mg1$
  12:u=$\underset{v\in V\setminus S}{\operatorname{argmax}}(v.mg)$
  13:S=S+{u}
}//While
```

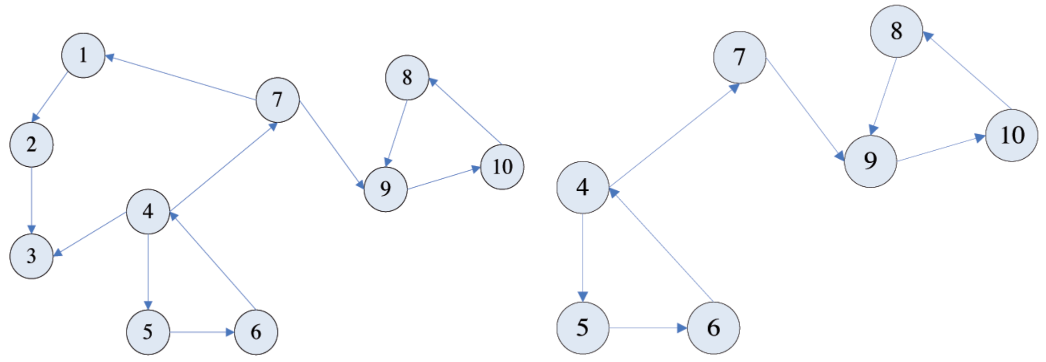
Example. We consider the graph G_i shown in Fig. 1(a), where V_i= {1,2,3,4,5,6,7,8,9,10}. We set S={1}. In line 6, we can get $F_{Gi}(S)$= {1,2,3}; in line 7, we can get induced graph G_i^s shown in Fig. 1(b). In line 8, we can get strong connected components, SCC_1= {4,5,6}, SCC_2= {7}, SCC_3= {8, 9, 10}, which is shown in Fig. 2. Suppose thresholdθ is 0.5, so, in line 9, we can get the macro graph SCC_i^s , which is shown in Fig. 3 . And, get the reachable set of SCC_1,SCC_2 and SCC_3, that is {SCC1,SCC2,SCC3}, {SCC2,SCC3} and {SCC3}. In line 10, we can get MG(G_i,S,v) for all nodes in G_i^s, that is, 1.mg1=2.mg1=3.mg1=0,4.mg1=5.mg1=6.mg1=|SCC1|+|SCC2|+|SCC3|=3+1+3=7,7.mg1=|SCC2|+|SCC3|=1+3=4,8.mg1=9.mg1=10.mg1=|SCC3|=3.

In algorithm2, u.mg1 is used to store the marginal gain of u for the current iteration, and u.mg is used to get the expected marginal gain of u for all iterations.

4.2.2 Reducing the number of calling influence function

From the algorithm KK-Greedy, we know that when we choose a new seed, any node in V\S as a candidate node would be examined by running influence function. So, it will reduce the efficiency of the algorithm. With the submodularity of influence function, CELF algorithm was proposed [11]. In my opinion, when a new seed u is selected, any node v in the marginal gain of influenced set should not be as a candidate node. Since v can be influenced by seed set S+{u}, all nodes that can be influenced by v can be influenced by seed set S+{u} too. So, if v will be chosen as the next seed, its marginal gain will be zero. Based on this idea, we proposed an improved algorithm LTN_CELF for CELF.

In LTN_CELF, we maintain a table Q<u,u.mg,mgset,u.flag> for all candidate nodes. In Q, u.mgset= getPosInfluenceSet(g,S+{u},q)-getPosInfluenceSet(g,S,q), u.mg= |u.mgset|, and u.flag is the number of iteration when u.mg was last updated. In our algorithm, when a new seed u is chosen, any nodes in u.mgset will be removed from Q, which is different from algorithm CELF. LTN_CELF is detailed in Algorithm3.



(a) graph G_i

(b)induced graph G_i^S (S={1})

Fig.1. An example of graph Gi and its induced graph

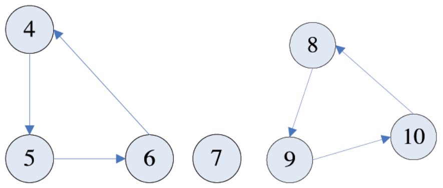


Fig.2. All strong connected components in G_i^S

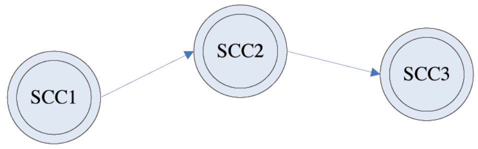


Fig.3. Macro graph SCC_i^S

Algorithm3 LTN_CELF _Greedy

Input: Social network G, Size of seed set K, Quality factor q

Output: Seed set S

```
1: S=ϕ; Q=ϕ;
2:For each v in V do {
3:    u.mgset=getPosInfluenceSet(G,{v},q)
4:    u.mg=|u.mgset|
5:    u.flag=0
6:    add u to Q by u.mg in descending order    }//For
7:While |S|<k  and |Q|>0 do  {
8:    u=Q[top]
9:    if u.flag==|S| then  {
10:     S=S+{u}
11:     Q=Q-u.mgset  }
12: Else {
13:   u.mgset= getPosInfluenceSet (G,S+{v},q)- getPosInfluenceSet (G,S,q)
14:   u.mg=|u.mgset|
15:   u.flag=|S|
16:   Resort Q by u.mg in descending order }
17:}//While
```

4.2.3 A mixed greedy algorithm for influence maximization of the LTN model

To improve the efficiency of influence function, an efficient algorithm LTN_NewGreedy is proposed in this study. To reduce the number of calling influence function, an improved algorithm LTN_CELF for algorithm CELF is proposed. Based on the idea of MixedGreedy [8], algorithm LTN_MixedGreedy is proposed. In the first iteration, the method in LTN_NewGreedy is used to get u.mg for all u in V, then, the method in LTN_CELF is used to reduce the times of calling influence function. Algorithm LTN_MixedGreedy is described in algorithm 4.

Same to algorithm LTN_CELF, the algorithm maintains a table Q<u,u.mg1,u.mg, u.mgset,u.flag> for every node u in V. In Q, u.mgset = getPosInfluenceSet(g,S+{u},q)-getPosInfluenceSet(g,S,q), u.mg1=|u.mgset|, and u.mg is the average marginal gain of node u.

Algorithm4 LTN_MixedGreedy

```
Input: Social network G, Size of seed set K, Quality factor q, threshold θ
Output: Seed set S
/* initialise */
1:S=ϕ;Q=ϕ;
2:For u in V do {
3:   u.mg1=0; u.mg=0; u.mgset={};u.flag=0;
4:   Add u to Q   }
/* get u.mg for all u in V  with the method used in algorithm LTN_NewGreedy*/
5:For v in V do {
6:  For i=1 to R do:{
7:    Get MG($G_i$,S,v) with formula (4), v.mg1= MG($G_i$,S,v) }//for
8:  Get the average marginal gain of node v for R iterations, v.mg=(1/R)*$\sum_{i=1}^{R} v.mg1$
9:}//for
10:Resort Q by v.mg in descending order
/*Get the seed set with the method used in algorithm LTN_CELF_Greedy
11:While |S|<k  and |Q|>0 do {
12:   u=Q[top]
13:   if u.flag==|S| then {
14:      S=S+{u}
15:      Q=Q-u.mgset }
16:  Else {
17:        u.mgset= getPosInfluenceSet (G,S+{v},q)- getPosInfluenceSet (G,S,q)
18:      u.mg=|u.mgset|
19:      u.flag=|S|
20:      Resort Q by u.mg in descending order  }
21:} //While
```

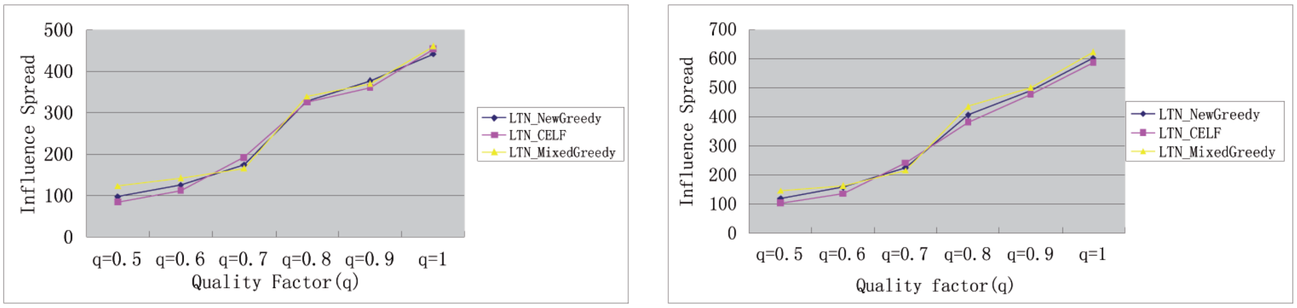
As described above, in lien 1-4, the algorithm initialize <u,u.mg1,u.mg, u.mgset,u.flag> for every node u in V.Then in line 5-9, the algorithm gets the average marginal gain for every node v in V with the method employed in altorithm LTN_NewGreedy. Comparing with algorithm LTN_CELF, the computation of v.mg with the same method used in algorithm LTN_NewGreedy.Then, in line 11-21, with the method used in algorithm LTN_CELF, the algorithm can get the seed set.

5. EXPERIMENTS

We implement algorithms NewGreedy, LTN_NewGreedy, CELF, LTN_CELF_Greedy, MixedGreedy, LTN_MixedGreedy, and conduct them on two real-world networks. We are interested in comparing both the influence spread and the running time of these algorithms.

5.1 DATASET

To evaluate the algorithms proposed in this work, two real-world datasets Epinions and Slashdot have been used. The two



(a)Epinions

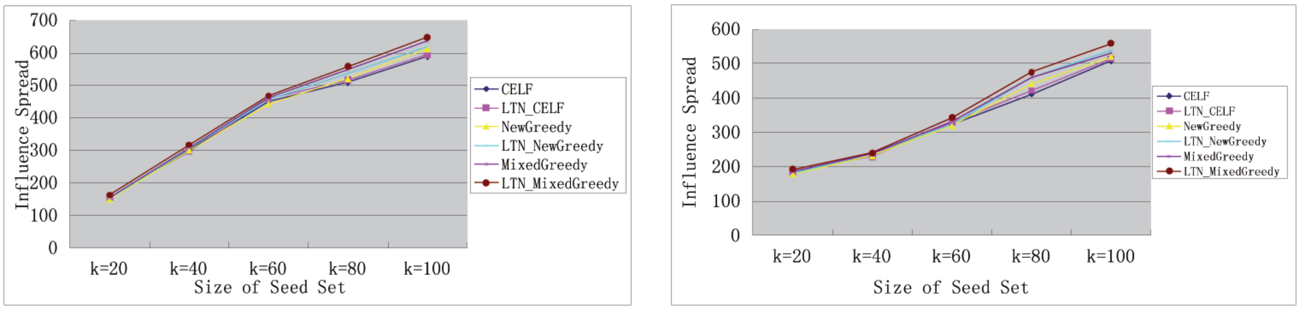
(b) Slashdot

Fig. 4 Influence spread vs. q on on Epinions and Slashdot (k=50)

id	dataset	nodes	edges	Average clustering coefficient	Number of SCC
1	Epinions	75,879	508,837	0.1378	42176
2	Slashdot	82,168	948,464	0.0603	10559

Table 2 Statics of datasets

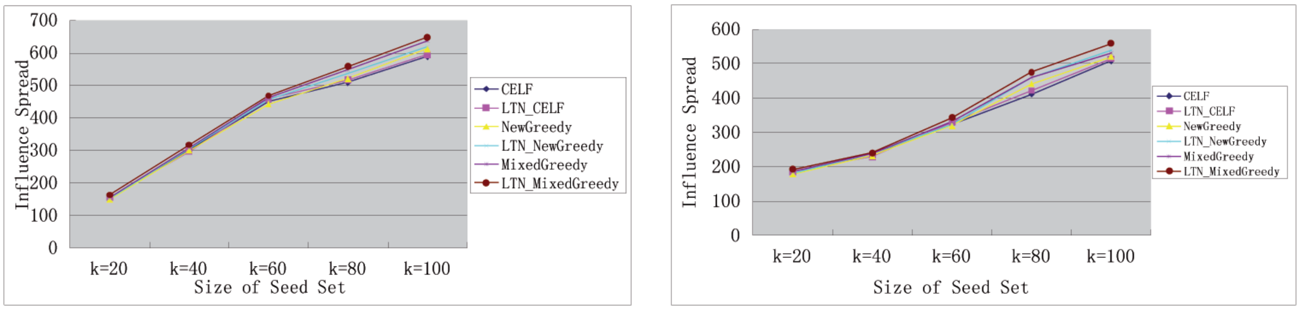
datasets were both collected from Stanford Large Network Dataset Collection (http://snap.stanford.edu /data/index.html). Epinions is a who-trust-whom social network (http://www.epinions.com). All the trust relationships interact and form the web of trust. The dateset Epinions was cited in [23]. Slashdot is a technology-related news website know for its specific user community. The dataset Slashdot contains friend/foe links between the users of Slashdot. The dataset Slashdot was cited in [24]. Basic statistics about these two networks are given in Table 2. For the weight of every edge, firstly, we assign it a random value in [0,1], then, we normalize it by $w_{ij}=w_{ij}/\sum_i W_{ij}$ .For the algorithm LTN_NewGreedy and LTN_MixedGreedy, the parameter θ is 0.5. In all experiments, the number of simulations R is 10000. The experiments are runing on a desktop computer with I5 2400S and 4G memory.



(a)Epinions

(b) Slashdot

Fig. 5 Influence spread vs. k on Epinions and Slashdot (q=0.8)



(a)Epinions

(b) Slashdot

Fig. 6 Running time vs. k on Epinions and Slashdot (q=0.8)

5.2 EXPERIMENTAL RESULTS

5.2.1 Quality factor on influence spread

First, we run algorithm LTN_NewGreedy,LTN_CELF and LTN_MixedGreedy on Epinions and Slashdot to get a 50-node seed set, with the quality factor q from 0.5 to 1.The results of this experiment are shown in Fig. 4. From Fig. 4, we can see that, when q increases, the influence spread increases quickly. The reason is that, if the quality of the product drops, the negative opinion would be more dominant. Therefore, the result suggests that maintaining a high product quality is very important in achieving a high influence spread.

5.2.2 Positive influence spread and running time on two datasets

In order to evaluate the performance of these algorithms, we run algorithms NewGreedy, LTN_NewGreedy, CELF, LTN_CELF_Greedy, MixedGreedy, LTN_MixedGreedy on Epinions and Slashdot. In our experiments, we set q=0.8. The influence spread of these algorithms on Epinions and Slashdot are shown in Fig. 5. And the running time of these algorithms on data1 and data2 are shown in Fig. 6.

As for the influence spread, from Fig. 5, we can see that the influence spread of these algorithms is matching. As for the running time, from Fig. 6, we can see that, algorithm LTN_CELF is faster than CELF; algorithm LTN_NewGreedy is faster than NewGreedy; algorithm LTN_MixedGreedy is faster than MixedGreedy. These results are expected. In these algorithms, LTN_MixedGreedy combines all advantages of LTN_CELF and LTN_NewGreedy, and it has the highest efficiency. In summary, in our experiments, LTN_MixedGreedy combines all advantages of LTN_CELF and LTN_NewGreedy, and has the highest efficiency in all algorithms for influence maximization.

6. CONCLUSION

Incorporating the propagation of negative opinions in viral marketing, based on the LT model, with a parameter quality factor q, we propose an extended model LTN for the LT model. For the LTN model, we propose some good properties, such as monotonicity and submodularity. As for the influence maximization of the LTN model, we give a simple approximate algorithm KK-Greedy with a ratio of (1-1/e). To improve the efficiency of the influence function getPosInfluenceSet(G,S,q), with the strong connected components in a graph, we propose an improved algorithm LTN_NewGreedy. To reduce the number of calling influence function, based on CELF algorithm, with the elimination of redundant candidate nodes, we propose an improved algorithm LTN_CELF algorithm for CELF. Based on the idea of MixedGreedy, combining the advantages of LTN_NewGreedy and LTN_CELF, we propose a new algorithm LTN_MixedGreedy. With experiments on two real-world datasets, we show that our improved algorithms have matching influence with their original algorithms, while being faster.

APPRECIATION

This work was supported in part by the National Foundation Science (No. 60673136,No. 61472340) and Hebei province colleges and universities science and technology research for Youth Fund Project(No. QN2014083)

BIBLIOGRAPHY

[1] Granovetter M. "Threshold models of collective behavior".American journal of sociology.May 1978. Vol. 83-6.p.1420-1443.DOI: http://dx.doi.org/10.1086/226707

[2] Watts D J. "A simple model of global cascades on random networks". in Proceedings of the National Academy of Sciences, 2002. April 2002.p. 5766-5771.DOI: http://dx.doi.org/10.1073/pnas.082090499

[3] Chen W, Collins A, Cummings R, et al. "Influence maximization in social networks when negative opinions may emerge and propagate". in Proceedings of the SIAM International Conference on Data Mining, 2011.April 2011.p.379-390.DOI: http://dx.doi.org/10.1137/1.9781611972818.33

[4] Domingos P, Richardson M. "Mining the network value of customers". in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining ,2001.August 2001.p.57-66. DOI: http://dx.doi.org/10.1145/502512. 502525

[5] Richardson M, Domingos P. "Mining knowledge-sharing sites for viral marketing",in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining ,2002. July 2002.p.61-70.DOI: http://dx.doi.org/10.1145/775047.775057

[6] Kempe D, Kleinberg J, Tardos É. "Maximizing the spread of influence through a social network". in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining ,2003. August 2003. p.137-146.DOI: http://dx.doi.org/10.1145/956750.956769

[7] Wang C, Chen W, Wang Y. "Scalable influence maximization for independent cascade model in large-scale social networks". Data Mining and Knowledge Discovery.Vol 25-3. November 2012.p.545-576.DOI: http://dx.doi.org/10.1007/s10618-012-0262-1

[8] Chen W, Wang Y , Yang S. "Efficient influence maximization in social networks" .in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining , 2009. June 2009. p.199-208.DOI: http://dx.doi.org/10.1145/1557019.1557047

[9] Chen W , Yuan Y , Zhang L."Scalable influence maximization in social networks under the linear threshold model". in proceedings of the IEEE 10th International Conference on Data Mining(ICDM), 2010.December 2010.p.88-97.DOI: http://dx.doi.org/10.1109/ICDM.2010.118

[10] Narayanam R, Narahari Y."A shapley value-based approach to discover influential nodes in social networks". IEEE Transactions on Automation Science and Engineering.Vol. 8-1. January 2011.p.130-147.DOI: http://dx.doi.org/10.1109/TASE.2010.205 2042

[11] Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N. " Cost-effective outbreak detection in networks". in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007. August 2007. p.420-429.DOI: http://dx.doi.org/10.1145/1281192.1281239

[12] Goyal A, Lu W, Lakshmanan L V. "Celf++: optimizing the greedy algorithm for influence maximization in social networks ". in Proceedings of the 20th international conference companion on World wide web, 2011. March 2011. p.47-48. DOI: http://dx.doi.org/10.1145/1963192.1963217

[13] Kimura M , Saito K. "Tractable models for information diffusion in social networks ". in Proceedings of the Knowledge Discovery in Databases: PKDD 2006. September 2006. p.259-271. DOI: http://dx.doi.org/10.1007/11871637_27

[14 Wang Y , Cong G , Song G , Xie K. "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks" in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. July 2010. p.1039-1048. DOI: http://dx.doi.org/10.1145/1835804.1835935

[15] Chen Y, Peng W,et al. "Efficient algorithms for influence maximization in social networks". Knowledge information systems. December 2012.Vol. 33-3 .p.577-601.DOI: http://dx.doi.org/10.1007/s10115-012-0540-7

[16] Hu J, Meng K, Chen X, et al. "Analysis of influence maximization in large-scale social networks". ACM SIGMETRICS Performance Evaluation Review. April 2014.Vol. 41-4.p.78-81.DOI: http://dx.doi.org/10.1145/2627534

[17] Lee J, Chung C. "A Query Approach for Influence Maximization on Specific Users in Social Networks". IEEE Transactions on knowledge and data engineering. February 2015.Vol. 27-2. P.340-353. DOI: http://dx.doi.org/10.1109/TKDE.2014.2330833

[18] Lu Z, Zhang W, Wu W, et al. "The complexity of influence maximization problem in the deterministic linear threshold model". Journal of combinatorial optimization.Vol 24-3. December 2012 .p.374-378. DOI: http://dx.doi.org/10.1007/s10115-012-0540-7

[19] Bharathi S, Kempe D, Salek M. "Competitive influence maximization in social networks". Internet and Network Economics, Berlin Heidelberg :Springer. December 2007. p.306-311. DOI: http://dx.doi.org/10.1007/978-3-540-77105-0_31

[20] Nazemian A, Taghiyareh F. "Influence maximization in Independent Cascade model with positive and negative word of mouth". in proceedings of the IEEE Sixth International Symposium on Telecommunications(IST) . November 2012. p.854-860.DOI: http://dx.doi.org/10.1109/ISTEL.2012.6483105

[21] Rozin P, Royzman E B. "Negativity bias, negativity dominance, and contagion".Personality and social psychology review. November 2001. Vol. 5-4. p.296-320. DOI: http://dx.doi.org/10.1207/S15327957PSPR0504_2

[22] Tarjan R . "Depth-first search and linear graph algorithms". SIAM journal on computing. June 1972. Vol. 1-2. p.146-160. DOI: http://dx.doi.org/10.1137/0201010

[23]]M. Richardson and R. Agrawal and P. Domingos. "Trust Management for the Semantic Web". The Semantic Web - ISWC, 2003. p.351-368.DOI: http://dx.doi.org/10.1007/978-3-540-39718-2_23

[24] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. "Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters". Internet Mathematics. January 2009. Vol. 6-1.p.29-123, DOI: http://dx.doi.org/10.1080/15427951.2009.10129177

[25] Min, Yang, Li Qianmu, and Song Yaoliang. "A SEIR Model Epidemic of Virus on the Online Social Network.". Journal of Digital Information Management. April 2014. Vol. 12-2.p.103-107.