Influence maximization by leveraging the crowdsensing data in information diffusion network

Feng Wang, Wenjun Jiang, Guojun Wang, Song Guo

PII: S1084-8045(19)30093-1
DOI: https://doi.org/10.1016/j.jnca.2019.03.002
Reference: YJNCA 2338

To appear in: Journal of Network and Computer Applications

Received Date: 23 October 2018
Revised Date: 24 January 2019
Accepted Date: 4 March 2019


This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
Influence Maximization by Leveraging the Crowdsensing Data in Information Diffusion Network

Feng Wang\textsuperscript{a,b}, Wenjun Jiang\textsuperscript{c}, Guojun Wang\textsuperscript{d,*}, Song Guo\textsuperscript{e}

\textsuperscript{a}School of Automation, China University of Geosciences, Wuhan 430074, China
\textsuperscript{b}Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China
\textsuperscript{c}College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
\textsuperscript{d}School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China
\textsuperscript{e}Department of Computing, Hong Kong Polytechnic University, Hong Kong, China

Abstract

The algorithm of influence maximization aims at detecting the top-k influential users (seed set) in the network, which has been proved that finding an optimal solution is NP hard. To address this challenge, finding the trade-off between the effectiveness and efficiency may be a more realistic approach. How to accurately calculate the influence probability is a fundamental and open problem in influence maximization. The existing researches mainly adopted the pair-wise parameters to denote the influence spread probability. These approaches suffer severe over-representing and overfitting problems, and thus perform poorly for the influence maximization problem. In this paper, we calculate the influence probability by learning low-dimensional vectors (i.e., influence vector and susceptibility vector) based on the crowdsensing data in the information diffusion network. With much fewer parameters and opposed to the pair-wise manner, our approach can overcome the overfitting problem, and provide a foundation for solving the problem effectively. Moreover, we propose the DiffusionDiscount algorithm based on the novel method of influence probability calculation and heuristic pruning approach, which can achieve high time efficiency. The experimental results show that our algorithm outperforms other five typical algorithms over the

*Corresponding author.
Email address: csgjwang@gmail.com (Guojun Wang)
real-world datasets, and can be more practical in large-scale data sets.

Keywords: Influence maximization, Low-dimensional vectors, Crowdsensing data, Greedy algorithm

1. Introduction

With the development of Online Social Networks (OSNs), the massive social media provides data foundation for online social network analysis. The wide spread researches in Influence Maximization (IM) pave the way for a large number of applications, such as viral marketing [1, 19, 20], outbreak detection [5], malware propagation prevention [26, 27], community detection [22], etc. The key issues in influence maximization is to find the top-\(k\) influential users (seed set) in the network. There are two major challenges in solving the problem of influence maximization. The first challenge is how to calculate the influence spread probability accurately, and the second one is how to design efficient algorithms that selects the top-\(k\) seed nodes.

Social influence occurs when a person’s emotions, opinions, or behaviors are affected by others [4, 28]. The data in the social network (social links, user attributes, etc.) cannot reflect users’ influence accurately. But the information diffusion data depict the information flows among individuals, which can denote the real data that change the behavior or opinion of others. Microblogs are our main source of the crowdsensing data. As illustrated in Figure 1, the attributes of microblog consist of the original content and comment. When an influenced user posts a report or expresses opinions to a post via comments, a social influence happens. Thus, the crowdsensing data in information diffusion network are normally publicly visible and can reflect the real influence between users.

To address the first challenge, the existing works mainly focuses on predicting the influence probability between each pair of users based on the data of social network [8-11, 23, 25], such as users’ demographic characteristics, the social network structure, etc. Almost all of these works simulate the influence probability in a pair-wise manner, i.e., using \(n^2\) independent parameters for \(n\) users. In addition, such pair-wise manner ignores dependency, and it is easy to cause over-fitting problem in the process of probability calculation, as exhibited in [6]. To better illustrate the problems above, we provide an intuitive example as shown in Figure 2.
Figure 1: The illustration of microblog.

(a) Over-representing problem  
(b) Over-fitting problem

Figure 2: The problem in the existing pair-wise influence works.

Example 1. Figure 2 (a) presents the over-representing problem in the pair-wise influence prediction method. To characterize the differentiated influence with each other for each pair of users, the existing works normally adopted the pair-wise parameters to denote the influence probability, i.e., the parameters of $p_{AC}$ and $p_{CA}$ denote the influence probability of user A to user C and user C to user A, respectively. One of the biggest problems is that there are too many parameters, resulting in that the influence maximization algorithm does not perform time efficiency in large-scale networks. In addition, it is easy to create over-fitting problem in the pair-wise influence prediction method, as shown in the Figure 2 (b). In such a case, the information initiated from user A, but the interpersonal influence for user D and user F cannot be learned. Because these two users do not appear in the same cascade. The pair-wise method will suppose the influence probability between nodes D and F to be zero. But in real life, it may have the indirect influence probability between
nodes D and F based on the extensive experimental experience. Moreover, power-law-like distributions are normally held in social network analysis, implying that the inter-influence of users with missing links cannot be ignored. Thus, the existing approaches are prone to overfitting the historical data, and the learning results are inconsistent with the actual situation. Therefore, this example shows that the existing methods based on the pair-wise parameters have serious over-representing and over-fitting problems. As opposed to the pair-wise manner, we propose to define parameters for individual users based on the low-dimensional vectors to effectively solve the problems above.

To address the second challenge, Kempe et al. [23] first formalized the influence maximization problem, and it is proved that the optimal solution for this problem is NP hard. To improve the efficiency of the general greedy algorithm, many algorithms have been proposed over the past years such as BKRIS [12], CELF [14], CELF++ [15], UBLF [16], etc., which are able to reduce the number of Monte-Carlo simulations (subgraph traversing). However, the existing approaches still suffer to the scalability issue over real-world data. Considering the characteristic of crowdsensing data, we propose to solve the influence maximization problem mainly based on the crowdsensing data in the information diffusion network. To the best of our knowledge, this may be the first attempt to use the low-dimensional vectors based on the crowdsensing data to explore the influence maximization problem.

This paper focuses on solving the problem of influence maximization based on the crowdsensing data in information diffusion network. First, we introduce the influence probability calculation method based on the low-dimensional vectors, specifically users’ influence vector and susceptibility vector, which denotes a unilateral influence from a user and a unilateral susceptibility of a user, respectively. The interpersonal influences are modeled by the product of the user’s susceptibility vector and the neighbor user’s influence vector. To calculate the influence probability under different topics, we need parameters in an order of $O(n \cdot i)$ to evaluate the influence probability under different topics, where $n$ and $i$ denote the number of users and topics, respectively. Moreover, we use the interpersonal influence to indicate the likelihood that information will be passed between two users. This method can overcome the problems of over-representing and overfitting. Second, we propose a seed selection algorithm, named DiffusionDiscount, based on the novel influence probability calculation method and heuristic pruning approach. The DiffusionDiscount algorithm evaluates the non-seed nodes’
marginal gain more efficiently. The experimental results illustrate our approach performs better than the existing works do. The main contributions of this paper are summarized as follows:

- We propose a novel method that simulates the influence spread process using much fewer parameters than the traditional methods to represent the influence probability. Our approach can overcome the problem of over-representing, and has advantage in modeling the influence spread process. Moreover, our approach can learn influence probability between users without overfitting.

- We calculate the influence probability based on the crowdsensing data in information diffusion network, to make use of the publicly visible data. Our approach iteratively selects the non-seed node with largest marginal gain without materializing all the samples, so it can accelerate the seed selection process, meanwhile it maintains the performance of influence spread.

- We conduct various experiments on the real-world datasets. We compare the influence spread range and running time with other five algorithms. The experimental results show the effectiveness and efficiency of our approach. Our DiffusionDiscount algorithm is more practical in large-scale datasets.

The rest of our paper is organized as follows. Section 2 reviews the related works. Section 3 provides the problem statement and approach framework. Section 4 introduces our influence probability calculation method based on the crowdsensing data. Section 5 presents the details of DiffusionDiscount algorithm. Section 6 evaluates the performance of influence maximization algorithms on the real-world data sets. Lastly, we conclude the paper in Section 7.

2. Related work

In this section, we briefly review the related works that best line up with our research, including influence probability calculation approaches and influence maximization algorithms.
2.1. Influence probability calculation

Many scholars focused on the problem of predicting the influence probability between users in online social networks [4, 8-13, 25]. The existing works normally assumed that the social edges between users labeled with influence probability. In recent years, several works focus on learning the influence probabilities from the real-world data [9, 25]. Wang et al. [7] simulated the influence spread process through fluid dynamics, then calculated the influence probability based on the novel influence spread model. Goyal et al. [8] proposed a novel algorithm to infer model parameters and applied the results to predict the influence spread. To avoid the costly Monte Carlo simulations for calculating influence probabilities, several works estimated influence spread based on the historical data [11, 12].

Although the existing works can measure and predict the influence probability, they are easy to cause the problems of over-representing and over-fitting, because they usually calculates the parameters in the paired form. However, these existing approaches are fundamentally different from the proposed framework in this paper, which focuses on the low-dimensional vectors (influence and susceptibility vector) based on the crowdsensing data in information diffusion network.

2.2. Influence maximization algorithm

In recent years, the existing works have extensively studied the Influence Maximization (IM) problem. There are two categories of typical approaches: heuristic algorithms [13] and greedy algorithms [14-16, 23].

To address the efficiency in influence maximization algorithm, heuristic algorithms have been proposed recently. The typical approach finds seeds through users’ degree. But heuristic algorithms usually perform much worse than the greedy algorithm does. The main reason is user’s influence value is determined by various factors, and we still do not know the evaluation factors exactly. Thus, we would like to propose a novel solution based on the greedy algorithm in this paper.

Kempe et al. [23] first formalized the influence maximization problem and presented a general greedy algorithm with the \((1 - 1/e - \epsilon)\)-approximation, where \(\epsilon\) denotes the error generated by using the Monte Carlo (MC) simulation. To address the efficiency of greedy algorithm, many algorithms have been proposed over the past years, such as BKRIS [12], CELF [14], CELF++ [15], UBLF [16], etc., which are able to reduce the number of MC simulation (subgraph traversing). Lucier et al. [17] presented a novel progressive
sampling algorithm, which can be used in the parallel framework to calculate the range of influence spread. Nguyen et al. [1] developed a novel approach that can reduce the sample size by using the stop-and-stage strategy. Lu et al. [2] analysed the bottleneck of the greedy algorithm, and propose a more efficient method to replace the time-consuming part of the greedy algorithm. Du et al. [5] formulated the IM problem as a submodular maximization task in a continuous-time diffusion model. However, the existing approaches still suffer on the scalability issue over the real-world data.

Li et al. [24] used mobile crowdsourced data obtained from location-based social network services to study influence maximization in LBSNs, but this work focused on the event activation position selection problem. Inspired by this work, we explore the influence maximization problem based on the crowdsensing data in information diffusion network.

3. Solution overview

3.1. Problem statement

The information diffusion network is essentially different from the traditional social network, although they may have several common links. The information diffusion network depicts the information flows among individuals while social network presents the social links among individuals. We obtain the calculation results of influence probability through the information flows. The node $E$ might include multiple information flows between one pair of nodes, which contributes to the influence probability learning. Therefore, due to the characteristic of crowdsensing data in information network, we inferring the influence probability based on the information diffusion network. And try to solve the influence maximization problem based on the novel influence probability calculation method.

Based on the crowdsensing data of information diffusion in a given time period, we build a information diffusion network $\mathcal{D}(V,E)$, where $V$ denotes all the individuals that occurring in the information diffusion process and $E$ denotes the diffusion edges between individuals.

Problem definition. Given an information diffusion network $\mathcal{D}(V,E)$ with the influence probability $p_{uv}$ between node $u$ and $v$ (low dimensional vectors), a positive integer $k$ and a certain propagation model, the influence maximization problem focus on selecting $k$ nodes as seed set $S \subseteq V$ that maximizes the influence spread.
Intuitively, we will select the top-\(k\) nodes with high value of influence as the seeds, but we have ignored the influence spread’s overlapping effect between seed nodes. Thus, we focus on the non-seed node’s marginal gain under the impact of current seeds, the calculation method of \(\delta(v, S)\) is shown as follows [23]. The function \(F(\cdot)\) denotes the influence spread range.

\[
\delta(v, S) = F(S \cup \{v\}) - F(S)
\]  

(1)

3.2. Approach framework

As shown in Figure 3, our framework consists of three parts, including influence probability calculation, initial influence evaluation and marginal gain calculation. First, we calculate the influence probability between neighbor nodes based on the crowdsensing data in information diffusion network. We obtain the results of influence probability through the low-dimensional vectors learning (influence vector and susceptibility vector). Second, we estimate each node’s initial influence value based on the PageRank algorithm, and conduct the social influence adjustment. Then we calculate non-seed node \(u\)’s influence loss on his neighbor node \(v\) as \(IL(u, v, P^S)\). Thus, we calculate node \(u\)’s marginal gain \((\delta(u, S))\) via deleting its impact on neighbors from their initial influence. Lastly, we can select node \(u\) as seed node if \(u = \arg \max_{v \in V \setminus S} \delta(v, S)\). Details of these calculation process are presented in the following sections.

4. Influence probability learning

To model the influence spread process accurately, the most important issue is inferring influence probability between individuals, which is fundamental to influence maximization. Different from the existing works, our
proposed approach calculates the influence probability based on the crowdsensing data of information diffusion. Specifically, we propose to model the interpersonal influence between any individuals through the influence and susceptibility vectors. In this section, we will introduce the influence and susceptibility vectors learning method, and the related parameters estimation and inference approach.

4.1. Influence and susceptibility vectors learning method

In this subsection, we present the influence and susceptibility vectors learning method based on the crowdsensing data in a given time period. With influence and susceptibility vectors of each individual, we can calculate the influence probability between neighbor nodes and supply basis to solve the problem of influence maximization.

Given a message \( m \), we denote its diffusion as \( D^m = (u, v, t_v) \), where each tuple \((u, v, t_v)\) denotes individual \( v \) forwards messages (or comments) at time \( t_v \) after getting it from individual \( u \). Considering different topics in the process of influence spread, we suppose each individual with two \( i \)-dimensional vectors: influence vector \( I_u \) and susceptibility vector \( S_u \). So the number of parameter in our method is \( O(n \cdot i) \). Influence vector denotes the intrinsic influence capability of individuals over others, and susceptibility vector reflects the strength that one individual is inclined to be influenced by others. When an individual \( v \) receives a message \( m \) at time \( t \), the influence probability \( p^m_v(t) \) that he will forward this message depends on the product of its active neighbor individuals’ influence vector and its own susceptibility vector. We use the parameter \( a^m_u(t) \) to denote whether individual \( u \) has forwarded message \( m \) before time \( t \), the parameter \( A \) denotes the activity level of overall users, and the computation method of influence probability \( p^m_v(t) \) is shown as follows.

\[
p^m_v(t) = \exp(A \cdot \sum_{u \in N(v), a^m_u(t) = 1} I_u^T S_v) \quad (2)
\]

Thus, \( 1 - p^m_v(t) \) means the probability that individual \( v \) refuse to forward message \( m \) at time \( t \). This is a well-defined Bernoulli probability distribution since the low-dimensional vectors are both nonnegative, and \( p^m_v(t) \in [0, 1] \).

Therefore, we can calculate the likelihood of observing the diffusion \( D^m = (u, v, t_v) \) of message \( m \) shown as follows.

\[
p(D^m | I, S) = \prod_{v \in V^m} p^m_v(t_v) \prod_{v \in D^m} (1 - p^m_v(\infty)) \quad (3)
\]
The parameter $V^m$ denotes all the individuals involving the diffusion of message $m$, and $D^m$ contains all the individuals who refuse to propagate message $m$. Accordingly, the likelihood of observing the diffusion of all messages is shown as follows. $N$ denotes the number of crowdsensing message.

$$p(D|I, S) = \prod_{m=1}^{N} p(D^m|I, S) \quad (4)$$

We can obtain the maximum likelihood estimation to the influence and susceptibility vector of each individual through maximizing $p(D|I, S)$. Since the number of parameter is very large, to overcome the overfitting problem, we adopt Gaussian priors for influence and susceptibility vectors.

$$p(I|\mu_I, \sigma^2_I) = \prod_{u=1}^{n} N(I_u|\mu_I, \sigma^2_I) \quad (5)$$

$$p(S|\mu_S, \sigma^2_S) = \prod_{v=1}^{n} N(S_v|\mu_S, \sigma^2_S) \quad (6)$$

where $I$ denotes the identity matrix with size $k \times k$. Therefore, the joint probability distribution can be calculated as follows.

$$P(D, I, S) = p(D|I, S) \times p(I|\mu_I, \sigma^2_I) \times p(S|\mu_S, \sigma^2_S) \quad (7)$$

**4.2. Parameter estimation and inference**

In this subsection, we apply the Projected Gradient (PG) approach to conduct parameter estimation and inference. First, we integrate two maximization objects as follows.

$$G(D, I, S) = P(D, I, S)^{\alpha} P(D, I, S)^{1-\alpha} \quad (8)$$

where $\alpha$ denotes the coefficient parameter and function $G(\cdot)$ denotes the global joint probability distribution. Then we can get the log of $G(D, I, S)$ as follows.

$$\log G(D, I, S) = \alpha P(D, I, S) + (1 - \alpha) P(D, I, S) \quad (9)$$

Maximizing the objective function $G(D, I, S)$ over the latent influence and susceptibility vectors with hyper-parameters equals to minimize the following objective function.

$$L(D, I, S) = -\alpha \log P(D, I, S) + (\alpha - 1) \log P(D, I, S) \quad (10)$$
Lastly, we can calculate the gradient $\partial L/\partial S_v$ and $\partial L/\partial I_u$, and update $I$ and $S$ matrices with PG method until maximum epoch is reached or gradients vanish. Then we can get the results of influence probability between nodes.

5. DiffusionDiscount algorithm

To solve the influence maximization problem, we propose the DiffusionDiscount algorithm to select the seed set based on the novel influence probability results in Section 4. Firstly, we estimate each user’s initial influence using the algorithm of PageRank. Then we calculate non-seed users’ marginal gain through deleting his impact on neighbors from their initial influence, inspired by the idea of heuristic pruning approaches. Thus, we can select the seed node through the results of marginal gain iteratively.

5.1. Influence evaluation

Based on the crowdsensing data in information diffusion network, we can obtain the results of influence probability through the low-dimensional vectors learning in Section 4. We need to select the top-$k$ influential nodes from the network that can achieve the largest range of influence spread. The major bottleneck in the previous works is how to calculate each non-seed node’s marginal gain $\delta(v)$ efficiently. The series of greedy algorithm evaluates $\delta(v)$ through traversing massive subgraphs during the MC process. Recently, several works try to avoid this problem and solve IM problem with lower computational complexity, such as DegreeDiscountIC algorithm [18]. The DegreeDiscountIC algorithm evaluated $\delta(v)$ through the calculation of non-seed neighbors’ expected number. This approach coincides with our main idea of influence probability calculation, so we combine this method with our low-dimensional vectors learning approach to design a novel influence maximization algorithm.

Under the impact of seed set $S$, we calculate each non-seed node $u$’s influence loss on its neighbor node $v$, which is represented by $IL(u, v, P^S)$ in our approach. In practical terms, $IL(u, v, P^S)$ indicates the influence spread overlapping between the non-seed node $u$ and seed set $S$. At beginning of the algorithm, we suppose seed set $S = \emptyset$. And we can obtain the influence probability between neighbor nodes, so we suppose we can calculate each node’s influence value $\sigma(u)$ before the seeds selection. Thus, we can obtain the result of marginal gain $\delta(u)$ through removing $IL(u, v, P^S)$ from $\sigma(u)$, and we name our approach DiffusionDiscount. Since our algorithm do not
have to traverse the generated subgraphs in the calculation of marginal gain, our algorithm can significantly improve the efficiency of seed set selection.

Firstly, we need to calculate the initial influence based on the results of influence probability. Inspired by the idea of PageRank algorithm, this paper adopts this idea in the estimation of initial influence to evaluate the importance of nodes [3]. The PageRank algorithm calculates each page’s PR value recursively through accumulating the fractions of its neighbor pages’ importance (neighbor quantity and quality). Intuitively, we can calculate each user’s initial influence value recursively that corresponding to the PageRank algorithm. According to the influence probability, each user’s initial influence value can be distributed to all his connecting users, and the influence value can be calculated as the sum of its neighbor’s influence value. Moreover, we consider the influence probability as the linking probability in this paper.

Note that our algorithm is based on the results of novel influence probability, all the neighbors of node \( u \) activate this node with the probability \( P_u \), the calculation method of probability \( P_u \) can be shown as follows.

\[
P_u = 1 - \prod_{v \in \mathcal{N}^{in}(u)} (1 - p_{vu}) \tag{11}
\]

According to the idea of PageRank, the parameter \( P_u \) represents node \( u \)'s influence ability that from its neighbor nodes. The parameter \( P_u \) denotes the influence ratio which allocated to the neighbors of node \( u \). The initial influence \( \sigma(u) \) can be allocated to each incoming neighbor node \( v \) based on the value of \( P_u w_{uv} \), the calculation method can be shown as follows.

\[
w_{uv} = \frac{p_{vu}}{\sum_{s \in \mathcal{N}^{in}(u)} p_{su}} \tag{12}
\]

\[
\sigma(u) = \sum_{v \in \mathcal{N}^{out}(u)} P_u w_{uv} \sigma(v) \tag{13}
\]

After node \( u \) allocates initial influence to its neighbor nodes, it has the probability \( (1 - P_u) \) to keep as the initial influence. Then the initial influence can be modifying with the factor of probability \( (1 - P_u) \), and the final value of influence (after the \( i \)th iteration) can be calculated as follows.

\[
\sigma(u)^{(i)} = (1 - P_u)\sigma(u)^{(i-1)} + \sum_{v \in \mathcal{N}^{out}(u)} P_u w_{uv} \sigma(v)^{(i-1)} \tag{14}
\]
Corresponding to the PageRank algorithm [2, 3], the initial influence
equation can also be a system of linear equation
\[ y = ax + b \]. If \( \lim_{t \to \infty} a^t = 0 \),
the initial influence equation becomes
\[ x = b + ab + a^2b + \cdots + a^tb \]. After the
tth iteration calculation of \( a^tb \), we estimate each node’s influence ability as:
\[
\sigma(u) = b + ab + a^2b + \cdots + a^tb
\]  
(15)

5.2. Marginal gain

Under the influence of current seed set \( S \), this paper calculates each non-
seed node’s marginal gain iteratively. We use \( \delta(u, S) \) to denote node \( u \)'s
marginal gain under the impact of seed set \( S \). We can add a non-seed node
\( u \in V \setminus S \) to the seed set that can maximize the marginal gain \( S \cup \{u\} \). For
each \( v \in N^{out}(u) \), our algorithm subtracts influence loss \( IL(u, v, P^S) \) from
\( \sigma(u) \) to get the results of \( \delta(u, S) \). In addition, we use the parameter \( \sigma(u, v) \)
to denote the influence strength of node \( u \) on \( v \), the calculation method can
be shown as follows.
\[
\sigma(u, v) = P_v w_{uv} \sigma(v)
\]  
(16)

During the computation process of \( IL(u, v, P^S) \), we follow the condition
that node \( u \)'s neighbors cannot be influenced since it becomes seed. If \( v \in S \),
then the computation method is shown as follows.
\[
IL(u, v, P^S) = \sigma(u, v)
\]  
(17)

To calculate node \( u \)'s influence loss on node \( v \), we should consider other
two situations when \( v \notin S \). Firstly, we consider the situation of node
\( u \in N^{in}(v) \), node \( u \) has an incoming link to node \( v \). When the node \( v \) can
be influenced by the current seed set \( S \) with the probability \( P^S_v \), node \( u \)'s
influence on node \( v \) can be reduced by \( \phi^S_v = P^S_v P_v \). Therefore, under this
situation, we can calculate node \( u \)' influence loss on node \( v \) as follows.
\[
IL(u, v, P^S) = \phi^S_v \sigma(u, v)
\]  
(18)

Secondly, we consider the situation of node \( u \in N^{out}(v) \), node \( v \) has an
incoming link to node \( u \). We can decrease the marginal gain \( \delta(u) \) by the
ratio of \( P^S_u \) when node \( u \) can be influenced by the current seed set \( S \) with
the probability \( P^S_u \). Therefore, under this situation, we can calculate node
\( u \)' influence loss on node \( v \) as follows.
\[
IL(u, v, P^S) = P^S_u \sigma(u, v)
\]  
(19)
Summarize the above considerations, we can obtain the results of marginal gain $\delta(u, S)$ through subtracting $IL(u, v, P^S)$ from node $u$’s influence $\sigma(u)$, which is shown as follows.

$$
\delta(u, S) = \sigma(u) - \sum_{v \in N^{out}(u)} IL(u, v, P^S)
$$

(20)

**Algorithm 1** The DiffusionDiscount algorithm $(G, P, k)$.

**Input:**
- Information diffusion network $D$
- Influence probability $P$
- Positive integer $k$

**Output:**
- Seed set $S$

1: Calculate the result of influence propagation probability between nodes based on the crowdsensing data in information diffusion network.

2: $S = \emptyset$

3: for round 1 to $k$

4: Evaluate each node’s initial influence: $\sigma^{(i)}(u) = (1 - P_u)\sigma^{(i-1)}(u) + \sum_{v \in N^{out}(u)} P_u w_{uv} \sigma^{(i-1)}(v)$.

5: for each non-seed node $u$

6: Calculate node $u$’s influence loss on its neighbor node $v$ as $IL(u, v, P^S)$:

7: if $v \in S$

8: $IL(u, v, P^S) = \sigma(u, v)$

9: if $v \notin S$

10: if $u \in N^{in}(v)$

11: $IL(u, v, P^S) = \varphi^S_{\sigma}(u, v)$

12: else if $u \in N^{out}(v)$

13: $IL(u, v, P^S) = P^S_u \sigma(u, v)$

14: Calculate node $u$’s marginal gain $\delta(u, S)$:

15: $\delta(u, S) = \sigma(u) - \sum_{v \in N^{out}(u)} IL(u, v, P^S)$

16: Select node $u \notin S$ as seed node

17: if $u = \arg \max_{v \in V\setminus S} \delta(v, S)$.

18: return the seed set $S$.

The details of DiffusionDiscount algorithm are shown in Algorithm 1. First, we calculate the influence probability between neighbor nodes based
on the crowdsensing data in information diffusion network (line 1). Second, we calculate each node’s initial influence value (line 4). Then we calculate influence loss of node $u$ on its neighbor node $v$ as $IL(u, v, P^S)$ (line 7-13). Thus, we can calculate node $u$’s marginal gain $\delta(u, S)$ in line 14-15. Lastly, we can select node $u$ as seed node if $u = \arg \max_{v \in V \setminus S} \delta(v, S)$.

**Time complexity.** At first, we should evaluate the influence probability $P^S$, the time complexity of this step is $O(|D||E|)$, where $|D|$ is the number of diffusion information and $|E|$ means the number of link edges between users. Then we need to calculate each node’s initial influence in the DiffusionDiscount algorithm. In the third step, we can calculate the influence loss $IL(u, v, P^S)$, the time complexity is $O(|E|)$. After this, we can calculate each non-seed nodes’ marginal gain according to the line 15 in Algorithm 1, the time complexity of this step is $O(|E|)$. Lastly, to select the top-$k$ seeds from round 1 to $k$, the cost of this process is $O(k|E|)$. Thus, the total time complexity of our algorithm is $O(k|D||E|)$.

6. Performance evaluation

In this section, we evaluate the performance of influence maximization algorithms on the real-world data sets. We implement all algorithms using Python in this paper, and run the algorithms on a 8.00 Gb memory, 3.20 GHz quad core Intel i7 CPU machine.

6.1. Data

We conduct various experiments on the real-world data of Sina Weibo, which is the largest social media in China. Using the information Application Programming Interface (API) on Sina Weibo open platform\(^1\), we crawl the crowdsensing data of information diffusion. The crowdsensing data contains the microblogs in a period of time (January 3, 2014 to May 12, 2014). Each microblog mainly contains the published content and time, ID number of blog, and ID number of user, etc. Moreover, we collect each user’s profile information which includes the microblogs, comments, retweets, relations between other users. Our dataset contains 63641 users with 84168 posts, 27759 retweets, 235183 comments, and the number of link between users is 1391718. This paper mainly focuses on four topics, including Fog and haze,

\(^1\)http://open.weibo.com/
Housing price, Xiaomi cellphone, and Public servant. The average length of diffusion episodes is 23.1. The distribution of users’ degree fits with the power-law distribution as shown in Figure 4.

In the part of case study, we conduct the experiments of performance evaluation on other three real-world data sets, which are widely used in the influence spread analysis. (1) **Epinions.** This is a who-trust-whom OSNs of a general consumer review site Epinions.com [21]. (2) **WikiVote.** This is a network for voting in Wikipedia, where nodes in the network represent Wikipedia users [7]. (3) **NetHEPT.** This network extracted from High Energy Physics - Theory section of the e-print arXiv [20]. We summary their basic statistics in Table 1.

**Table 1:** Statistics of the real-world data sets

<table>
<thead>
<tr>
<th>Data</th>
<th>Sina Weibo</th>
<th>Epinions</th>
<th>WikiVote</th>
<th>NetHEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>63641</td>
<td>131828</td>
<td>7115</td>
<td>27770</td>
</tr>
<tr>
<td>Edges</td>
<td>1391718</td>
<td>841372</td>
<td>103689</td>
<td>352807</td>
</tr>
<tr>
<td>Avg. Degree</td>
<td>12.9</td>
<td>6.1</td>
<td>14.5</td>
<td>12.2</td>
</tr>
<tr>
<td>Max. Degree</td>
<td>70</td>
<td>3478</td>
<td>457</td>
<td>2414</td>
</tr>
</tbody>
</table>

Based on the crowdsensing data, we apply the triples forms (user, topic, timestamp) to denote the information diffusion process. The topic is extracted from the papers, microblogs, reposts, comments of users, etc. Each triplet indicates that at a known timestamp, a given user interacted with other users on the topic. Each topic can correspond to the diffusion process, which infects each user who interacted with it.
6.2. Experiment setup

We consider two classic influence spread models in our comparison experiments, including the Independent Cascade (IC) mode and Linear Threshold (LT) model [23]. We compare our DiffusionDiscount algorithm with other five algorithms. (1) Random algorithm applies the random probability to select the seed node. (2) Degree algorithm is a typical heuristic algorithm which selects seeds based on users’ degree [13]. (3) Greedy algorithm iteratively chooses seeds based on the marginal gain [23]. (4) PageRank algorithm selects the seeds mainly based on the network structure and users’ attributes [3]. (5) IMRank algorithm is a hybrid algorithm which using a novel ranking strategy to accelerate the seeds selection process [21].

In Random algorithm, we set a random probability \( p_{uv} \in [0, 1] \) to represent the user’s influence ability. In Degree algorithm, we set the influence probability as \( p_{uv} = \frac{n_{in}}{ave} \), which \( n_{in} \) is the incoming number of users, \( ave \) is the average number of linking users. In Greedy algorithm, to estimate the influence probability accurately, we set the number of MC simulation \( R = 10000 \). We set other parameters according to these algorithms, respectively [3, 21].

In order to illustrate the performance of influence maximization algorithm, the range of influence spread and the time complexity of seed set selection are two major metrics. The existing works normally suppose each link \((u,v)\) labeled with the influence probability \( p_{uv} \). In linear threshold model, we suppose 0.35 to be the threshold of activation probability.

6.3. Experimental results

6.3.1. The range of influence spread.

We use this evaluation metric to compare the number of activated users that affected by the most influential users (seed set) after the influence propagation process. In the influence spread simulation process, we assume 10,000 simulations after selecting a new user add to the seed set. Then we suppose the newly range of influence spread as the updated seeds’ influence ability. Therefore, we compare the influence ability of seed set that selected through different algorithm. We conduct experiments under various conditions, such as the seed size \( k \) varies from 5 to 55, and using different influence spread model (IC and LT model).

As shown in Figure 5 and 6, the experiments of influence spread are conducted on different seed size \( k \), topic and influence spread model, respectively.
In general, Greedy algorithm performs best under IC and LT model, which serves as a base for most of existing IM algorithms. The DiffusionDiscount (DD) algorithm outperforms other four algorithms, and maintains a close performance to Greedy algorithm under four different topics.

Figure 5 illustrates four different topics of influence spread under IC model. Under the topic of fog and haze and housing price, the DiffusionDiscount algorithm performs much better than Degree and PageRank do, and performs about 5.6% better than IMRank does. Under the topic of Xiaomi cellphone and public servant, PageRank performs better under the other two topics. Under the topic of Xiaomi cellphone, the DiffusionDiscount algorithm performs about 48% better than PageRank does, and 3.3% better than IMRank does. Under the topic of public servant, the DiffusionDiscount algorithm performs 42% better than PageRank and IMRank do.

Figure 6 shows four different topics of influence spread under LT model. Under the topic of fog and haze, Xiaomi cellphone and public servant, the DiffusionDiscount algorithm performs much better than Degree and PageR-
Figure 6: A comparison of influence spread of different topics under LT model

ank do, and performs about 3.1% better than IMRank does. Under the topic of housing price, Degree algorithm performs better than PageRank does. Note that the performance of PageRank and Degree algorithms vary greatly and could be affected by the structures of social network. The performance of DiffusionDiscount and IMRank are much more stable.

As shown in Figure 7, we compare the DiffusionDiscount algorithm’s influence spread performance based on different number of crowdsensing message. We conduct this experiment under the condition of seed set size $k = 55$ and four different topics in IC model. With the increasing number of crowdsensing message, the performance of influence spread becomes better. These results indicate that more crowdsensing data of information diffusion can improve the performance of DiffusionDiscount algorithm.

6.3.2. Running time.

We compare the time complexity of seed set selection between six influence maximization algorithms. In this part of experiment, we select experiment conditions of seed set size $k = 55$ and three different topics (Fog
and haze, Housing price and Xiaomi cellphone). In addition, we conduct the experiments under the IC and LT influence spread model, respectively.

Table 2: Running time under IC model

<table>
<thead>
<tr>
<th>Topic</th>
<th>Random</th>
<th>Greedy</th>
<th>Degree</th>
<th>PageRank</th>
<th>IMRank</th>
<th>DiffusionDiscount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fog and haze</td>
<td>4.2s</td>
<td>N/A</td>
<td>12s</td>
<td>281s</td>
<td>180s</td>
<td>102s</td>
</tr>
<tr>
<td>Housing price</td>
<td>&lt;1s</td>
<td>N/A</td>
<td>21s</td>
<td>42s</td>
<td>55s</td>
<td>38s</td>
</tr>
<tr>
<td>Xiaomi cellphone</td>
<td>5.7s</td>
<td>N/A</td>
<td>37s</td>
<td>343s</td>
<td>308s</td>
<td>319s</td>
</tr>
</tbody>
</table>

Table 3: Running time under LT model

<table>
<thead>
<tr>
<th>Topic</th>
<th>Random</th>
<th>Greedy</th>
<th>Degree</th>
<th>PageRank</th>
<th>IMRank</th>
<th>DiffusionDiscount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fog and haze</td>
<td>4.8s</td>
<td>N/A</td>
<td>33s</td>
<td>292s</td>
<td>190s</td>
<td>129s</td>
</tr>
<tr>
<td>Housing price</td>
<td>2s</td>
<td>N/A</td>
<td>12s</td>
<td>82s</td>
<td>65s</td>
<td>51s</td>
</tr>
<tr>
<td>Xiaomi cellphone</td>
<td>4.5s</td>
<td>N/A</td>
<td>43s</td>
<td>355s</td>
<td>323s</td>
<td>303s</td>
</tr>
</tbody>
</table>

Table 2 and 3 show that different algorithms have huge difference in time complexity. We can neglect Random algorithm since the influence spread performance of Random algorithm is usually terrible. The running time of Degree algorithm is less than that of our DiffusionDiscount algorithm, but the performance of Degree algorithm is normally not good enough. The Greedy algorithm suffers on serious scalability issue because it has to select the seeds through the costly Monte Carlo simulations. The running time of Greedy algorithm can be nearly 28 hours. So Greedy algorithm has low availability in practical application even though this algorithm has the best performance of influence spread. The running time of DiffusionDiscount algorithm can be 315 times less than greedy algorithm’s running time. Moreover, the running
time of DiffusionDiscount algorithm is better than those of the PageRank and IMRank algorithms. In addition, our DiffusionDiscount algorithm can obtain the seed set within nearly 310 secs in large-scale dataset. Thus, the experimental results illustrate that our algorithm can make the trade-off between effectiveness and efficiency, and our algorithm is more practical in the large-scale data.

6.3.3. Case study.

In this subsection, we compare the performance of seed set selection between six influence maximization algorithms. In this part of experiment, we select experiment conditions of seed set size $k = 5-55$, and three different topics (The topic of Sony in Epinions, election campaigns in WikiVote and high energy physics in NetHEPT). In addition, we conduct the experiments under the IC and LT influence spread model, respectively.

![Graphs showing influence spread under IC model in different datasets.](image)

(a) Epinions  
(b) WikiVote  
(c) NetHEPT

Figure 8: A comparison of influence spread under IC model in different datasets.

As shown in Figure 8 and 9, although all the influence spread trends are ascending, each time the influence spread of DiffusionDiscount algorithm is greater than those of other four algorithms (Random, Degree, PageRank,
and IMRank), especially when the seed size becomes larger. This finding demonstrates the DiffusionDiscount algorithm can be better in the large social networks. The influence spread of DiffusionDiscount algorithm is much higher than those of Random, Degree and PageRank algorithms. The influence spread of DiffusionDiscount algorithm just a little worse than that of the Greedy algorithm, but we should make the trade-off between effectiveness and efficiency. The results of case study shows that the DiffusionDiscount algorithm can be more practical in the large-scale data.

7. Conclusion

In this paper, we propose a novel approach to calculate the influence probability using low-dimensional vectors based on the crowdsensing data in information diffusion network. Our approach needs much less parameters to represent the influence probability and can overcome the overfitting problem. Moreover, we propose the DiffusionDiscount algorithm based on the heuristic pruning approach that achieved high time efficiency, which can be 315 times faster than greedy algorithm in the real-world data set. Since our
approach do not need to traverse the generated subgraphs in the calculation of marginal gain, our algorithm can improve the efficiency of seeds selection, meanwhile it maintains the influence spread performance (compared to the greedy algorithm). We conduct extensive experiments on 4 different real social network dataset. The experimental results illustrate that based on the crowdsensing data in information diffusion network, 1) the DiffusionDiscount algorithm performs better than the baselines do in the real-world data; 2) the DiffusionDiscount algorithm can be more practical in large-scale data sets.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant Numbers 61632009, 61472451 and 61502161, the Guangdong Provincial Natural Science Foundation under Grant Number 2017A030308006 and the High Level Talents Program of Higher Education in Guangdong Province under Funding Support Number 2016ZJ01.

REFERENCES


Feng Wang received the PhD degree in computer science from Central South University, P. R. China, in 2018. He is currently a postdoctor at School of Automation, China University of Geosciences, Wuhan. His research interests include social network analysis, social influence evaluation models and algorithms, machine learning and recommendation systems.

Wenjun Jiang received the bachelor’s degree in computer science from Hunan University, P. R. China, in 2004, the master’s degree in computer software and theory from the Huazhong University of Science and Technology, P. R. China, in 2007, and the doctorate degree in computer software and theory from Central South University, P. R. China, in 2014. She has been a visiting PhD student at Temple University for two years. Currently, she is an assistant professor at Hunan University. Her research interests include trust and social influence evaluation models and algorithms in online social networks, recommendation systems, and social network analysis.

Guojun Wang received the B.Sc. degree in geophysics in 1992, the M.Sc. degree in computer science in 1996 and the Ph.D. degree in computer science in 2002, all from Central South University, Changsha, China. He is currently the Pearl River Scholarship Distinguished Professor at Guangzhou University, China. He was a Professor at Central South University, China; a visiting scholar at Temple University and Florida Atlantic University, USA; a visiting researcher at the University of Aizu, Japan, and a Research Fellow at Hong Kong Polytechnic University. His research interests include cloud computing, trusted computing, and information security. Dr. Wang is a distinguished member of the CCF, and a member of the ACM and IEICE.
Song Guo received the PhD degree in computer science from the University of Ottawa. He is currently a Full Professor at Department of Computing, the Hong Kong Polytechnic University. His research interests are mainly in the areas of wireless communication and mobile computing, cloud computing, big data, and cyber-physical systems. He has published more than 400 papers in refereed journals and conferences in these areas and received three IEEE/ACM best paper awards. He currently serves as Associate Editor of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Emerging Topics in Computing, and many other major journals. He has also been in organizing and technical committees of numerous international conferences. He is a senior member of the IEEE and the ACM.