Expert Systems With Applications 102 (2018) 70–82

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Identifying influential individuals in microblogging networks using graph partitioning

Mingqing Huang[a], Guobing Zou[a,*], Bofeng Zhang[a,*], Yanglan Gan[b], Susu Jiang[a], Keyuan Jiang[c]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[b] School of Computer Science and Technology, Donghua University, Shanghai 201620, China
[c] Department of Computer Information Technology and Graphics, Purdue University Northwest, Indiana 46323, USA

ARTICLE INFO

Article history:
Received 22 June 2017
Revised 12 February 2018
Accepted 13 February 2018
Available online 13 February 2018

Keywords:
Influential individual
Microblogging network
Graph partitioning
Latent Dirichlet Allocation model
LeaderRank

ABSTRACT

Identifying influential individuals who lead to faster and wider spreading of influence in social networks is of theoretical significance and practical value to either accelerating the speed of propagation in the case of product promotion, or hindering the pace of diffusion involved in rumors. Conventional methods, ranging from centrality indices to diffusion-based processes, already take into account the number and influences of followers, but fail to make full use of the characteristics of social media. A novel approach called PartitionRank for finding a pre-fixed number of influential individuals in microblogging scenarios is proposed in this study to maximize the impact; it combines interest similarity with social interaction between users via graph partitioning. Experimental results on artificial and real-world microblogging networks illustrate that our scheme outperforms the other state-of-the-art methods in effectiveness and efficiency.

1. Introduction

One main function of social networks is to propagate information, ideas, reputations, and influences between any two users (Kim & Song, 2011). The information can disseminate beyond the direct followers, occasionally passing to a mass of individuals. Information dissemination is a prevalent process that formally describes many dynamic network activities (Keeling & Rohani, 2008; Rogers, 1995). The knowledge of the roles that users play in the dissemination process is crucial for exploiting efficient methods to either accelerate or hinder dissemination.

It is a fundamental problem to mine a tiny fraction of influential individuals from a social network such that they can transfer information to the largest number of users (Kimura, Saito, Nakano et al., 2010; Kitsak, Gallos, Havlin et al., 2010). The solution to this problem has broad application prospects (Kaiser, Schlick, & Bodendorf, 2011; O'Mahony & Smyth, 2010). First, social media is a powerful marketing platform. Targeting influential individuals can improve the efficiency of marketing campaigns. Kempe et al. give a vivid instance (Kempe, Kleinberg, & Tardos, 2003): A company initially selects a small number of "influential" users in the social network by giving them free product samples and hopes that these users will recommend the product to their friends, and then influence their friends' friends; many users will thus finally purchase the new product through the powerful word-of-mouth effect. Second, there are many applications that utilize social media to collect opinions and information on special topics. Identifying influential individuals can significantly raise the quality of the collected opinions.

Designed to solve complex problems by reasoning about knowledge, after absorbing the technology of finding most influential people in social media, the expert systems have enlarged the applicable area, such as recommender systems development (Morid, Shajari, & Hashemi, 2014), useful weblogs choosing (Leskovec, Krause, Guestrin et al., 2007) and influential twitters identification (Weng, Lim, Jiang et al., 2010). For example, the intelligent medical auxiliary system can concretely estimate disease categories and accurately recommend suitable hospitals and experts according to the description of symptoms and the location coordinates of the patients.

Social networks exhibit the property of modular structure (Nguyen, Dinh, Xuan et al., 2011; Palla, Pollner, Barabási et al., 2009), i.e., they divide naturally into communities of vertices with denser connections inside each cluster and fewer connections crossing clusters, where vertices and connections express network

* Corresponding authors.
E-mail addresses: mqhuang@shu.edu.cn (M. Huang), gbzou@shu.edu.cn (G. Zou), bfzhang@shu.edu.cn (B. Zhang), ylgan@dhu.edu.cn (Y. Gan), susujiang@shu.edu.cn (S. Jiang), kjiang@pnw.edu (K. Jiang).

https://doi.org/10.1016/j.eswa.2018.02.021
0957-4174/

users and the social interactions among them, respectively. In this paper, we analyze the effects of some popular approaches on identifying influential individuals in social networks, including degree centrality, closeness centrality, betweenness centrality, and PageRank. Sometimes these approaches are limited when applied to social networks as they ignore the networks' community structure and do not identify the influential individuals from communities in a relatively balanced way. For instance, the individuals with a high degree that are often treated as influential roles may all lie in the same community with larger size such that they can only impact individuals in the same community (Zhang, Zhu, Wang et al., 2013). In this paper, we propose a novel approach to identify influential individuals in social networks.

Microblogging is a new representative form of communication. It allows users to release brief message updates (with a limit of 140 characters), which can be published in many different channels, including the Web and mobile phones.[1] Microblogging also provides "social-networking" functionality. Unlike other social networks that require users to grant permission to other users befriending them, microblogging adopts a social-networking model called "following", in which each user is allowed to choose who he/she wants to follow freely. Conversely, the user may also be followed by others without granting permission first. In an example of a "following" relationship, the user who is following is named the "follower", while the one whose updates are being followed is called the "friend". Microblogging has gained extensive popularity, and also has drawn huge interest from the research community.

The "following" relationship is a potential indicator of topic similarity among users (Weng et al., 2010). A user follows a friend because he/she is interested in the topics that the friend releases in microblogs, and the friend follows back because he/she finds that they share similar topic interests. This phenomenon is named "homophily", and it has emerged in many social networks (McPherson, Smith-Lovin, & Cook, 2001).

In this study, we measure the individual influence combining the "following" relationship and the topical similarity among users. First, the interest interaction network is constructed through topic distillation of microblogs. Second, all users in the network are divided into a pre-fixed number of communities in a relatively size-balanced way. Finally, the highest-ranked scorer in each community is returned as a small subset of influential individuals.

The main innovations and characteristics of this study are included as follows.

- An interest interaction network is built taking both interaction intimacy and interest similarity into account to depict the information spreading probabilities, which demonstrates as a directed and weighted multi-dimension network.
- Identifying influential individuals in social networks from the aspect of communities, which gives a particular distance from one another among spreading origins, can decrease and even dissipate the overlap.
- The modularity and size of communities are integrated to eliminate the negative effect in extremely unbalanced communities, which can further increase the spreading effectiveness.

The remainder of this paper is organized as follows. The related work on identifying influential individuals is summarized in Section 2. In Section 3, we elaborate on the proposed approach, which detects the most influential individual from each user group. The experimental evaluation on several computer-simulated and real-world networks is executed in Section 4. Finally, we conclude the paper and make suggestions for future research in Section 5.

2. Related work

It is well known that many mechanisms, such as spreading, cascading and synchronizing, are highly impacted by a small subset of influential individuals (Zamora-López, Zhou, & Kurths, 2010). How to identify these influential individuals is of theoretical significance and practical value. Moreover, detecting influential individuals is essential for design of effective information dissemination strategies in many fields, including rumor controlling, public health practices, business management, and marketing campaigns.

A variety of centrality indices have been proposed to solve this problem, such as degree centrality, closeness centrality (Sabidussi, 1966), betweenness centrality (Freeman, 1979), eigenvector centrality (Bonacich, 2007), *k*-shell decomposition (Kitsak et al., 2010), and local proxy (Pei, Muchnik, Andrade et al., 2014). Degree centrality is a simple and efficient metric, but it lacks relevance. For instance, an individual lying in the center of the network, which has a few highly influential followers, may be more influential than an individual existing at the periphery of the network and having a larger number of less influential followers. Closeness centrality, which can be referred to as a measure of how long it takes to spread information from an individual to all of the other individuals sequentially, may highlight the individuals located at the junction between communities. Betweenness centrality is defined as the fraction of the shortest paths between pairs that cross through the individual of interest. Individuals with high betweenness often act as intermediaries in transferring information, such that they play pivotal roles in information dissemination between communities rather than as initial spreaders, and they cannot satisfy the application requirements in large-scale social networks for their high computational complexity. Eigenvector centrality has limitations for directed and weighted networks of social media, since it only targets the undirected networks. The *k*-shell decomposition approach does not always work well, as sometimes the individuals in the core occupy a high proportion of the network, such that the influential individuals cannot be detected. Local proxy for individuals' influence – the sum of the nearest neighbors' degrees, can be further improved effectiveness by taking into account more factors, such as the clustering coefficient of nodes (Chen, Gao, Lü et al., 2013).

With the explosive growth of network data, a number of random-walk-based algorithms have been designed. The representative methods include the well-known PageRank (Brin & Page, 1998) and TunkRank,[2] as well as some recently proposed approaches, including LeaderRank (Li, Zhou, Lü et al., 2014; Lü, Zhang, Yeung et al., 2011) and TwitterRank (Weng et al., 2010). All of these algorithms assume that an individual is supposed to be of high influence if it is pointed to by many highly influential followers. It has been demonstrated that these approaches are superior to centrality-based methods in terms of ranking effectiveness. They may adapt to find the original influential promulgators only when the spreading originates in a single active individual. For a spreading process originating in many active individuals simultaneously, spreading origins located at a particular distance from each other must be confirmed, to avoid or reduce the repeated impact on many of the same individuals and extend the influence scope. However, these classic approaches may detect influential individuals who do not lie far enough away, since they do not take into account the community structure that is ubiquitous in social networks.

Inspired by humanities science, two well-known influence maximization models are suggested (Kempe et al., 2003) and they have been adopted to derive many different approaches, including

[1] http://en.wikipedia.org/wiki/Micro-blogging.

[2] http://thenoisychannel.com/2009/01/13/atwitter-analog-to-pagerank.

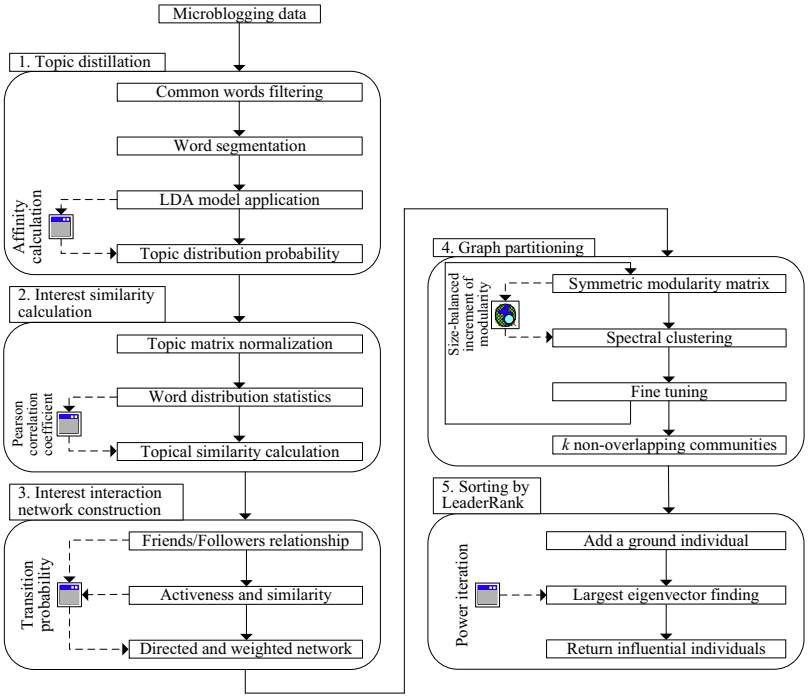


Fig. 1. System architecture of proposed approach.

the ComPath algorithm (Rahimkhani, Aleahmad, Rahgozar et al., 2015) based on the linear threshold model and a Monte-Carlo-simulation-based method (Ohsaka, Akiba, Yoshida et al., 2014) using the independent cascade model. The ComPath scheme provides a good balance between effectiveness and execution time to find the top-k most influential people in social networks. The Monte-Carlo-simulation-based algorithm exploits the existence of a hub in social networks to accelerate breadth-first searches for capturing solutions of high quality with a theoretical guarantee. All of these initiatives try to maximize the impact under stochastic simulation models, failing to combine the interest similarity between users for characterizing the influence spread in social networks.

3. Proposed approach

In this study, we select Chinese microblogging as the benchmark research platform. However, slight modification of the proposed approach can be adopted to any microblogging scenario in other languages, and even any social media.

3.1. Framework of proposed approach

For the purpose of clearly depicting the characteristics and content of the proposed approach, the detailed overall framework is displayed in Fig. 1. The proposed approach consists of five crucial steps, including topic distillation, interest similarity calculation, interest interaction network construction, graph partitioning, and sorting by LeaderRank, where the collection of microblogging data has been plugged in as the foundation. First, the topics of microblogs for each user are extracted using the Latent Dirichlet Allocation (LDA) model (Blei, Ng, & Jordan, 2003; Heinrich, 2008). Then, Pearson correlation coefficient is applied to calculate the interest similarity between each friend and follower pair. Next, we build a probability network for information transmission taking into account the social relationship, the interest similarity, and the user activity level. The directed and weighted network is then divided into a pre-fixed number of communities by the spectral clustering algorithm. Finally, the most influential user in each community is detected to form the influential individual set. In the following subsections, we describe each component in detail.

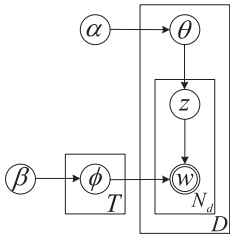


Fig. 2. Graphical illustration of LDA model.

3.2. Topic distillation

The purpose of topic distillation is to automatically identify the topics that users are interested in based on the microblogs they have released. The LDA model, an unsupervised machine learning technique to mine the latent topic information from a large document set, is adopted to achieve this goal. It treats each document as a "bag of words", so each document emerges as a probability distribution over some topics, and each topic emerges as a probability distribution over a lot of words. The generative process for each document is as follows.

(1) For each document, choose a topic from its distribution over topics.
(2) Take a word from the distribution over the words associated with the picked topic.
(3) Repeat steps (1) and (2) until all of the words in the document are sampled.

This generative process is graphically illustrated using universal plate notation in Fig. 2. Each of a collection of D documents is related to a multinomial distribution over T topics, which is represented as θ. Each topic is related to a multinomial distribution over a lot of words, represented as ϕ. θ and ϕ have Dirichlet priors

with hyper-parameters α and β, respectively. In this figure, dual-circular and single-circular plates denote observed and latent variables separately. A directed edge corresponds to a conditional dependency between two variables, and boxes denote repeated samplings, with the number of times given by the variable at the bottom right of the corresponding box. In addition, z is a topic taken from the multinomial distribution θ associated with the document, w denotes a word sampled from the multinomial distribution ϕ associated with the topic, and N_d stands for the number of words in the document.

In this study, Gibbs sampling is then adopted to estimate the model parameters from the data, i.e., the document-topic distributions θ and the T topic-word distributions ϕ.

The result is represented in two matrices:

(1) A $D \times T$ matrix, denoted DT, where D is the number of users and T is the number of topics. DT_{ij} represents the number of times a word in user s_i's microblogs has been assigned to topic t_j.
(2) A $W \times T$ matrix, denoted WT, where W is the number of unique words in all of the microblogs and T is the number of topics. WT_{ij} contains the number of times a unique word w_i has belonged to topic t_j.

Since the purpose is to identify the topics that each user is interested in rather than the topic that each single microblog is about, we assemble the microblogs released by the same user into a document, and then Chinese word segmentation[3] is executed for each document.

3.3. Interest similarity calculation

For the purpose of measuring interest similarity, matrix DT is then row-normalized as DT' such that $\|DT'_{i\cdot}\|_1 = 1$ for each row $DT'_{i\cdot}$. Each row of matrix DT' is essentially the probability distribution of user s_i's interest over the T topics, i.e., each element DT'_{ij} indicates the probability that user s_i is interested in topic t_j.

Several measurement criteria have been proposed in the literature for the sake of calculating the interest similarity between individuals, among which the Pearson correlation coefficient is most popular. Given this, a modified version of the Pearson correlation coefficient is employed to measure the interest similarity between users with a "following" relationship as follows.

Definition 1 (Interest similarity measure)**.** Interest similarity between two microblog users s_u and s_v can be calculated as

$$sim(u,v) = \frac{\sum_{t\in T} r_t (DT'_{ut} - \overline{DT'_u})(DT'_{vt} - \overline{DT'_v})}{\sqrt{\sum_{t\in T} r_t (DT'_{ut} - \overline{DT'_u})^2 \sum_{t\in T} r_t (DT'_{vt} - \overline{DT'_v})^2}}, \quad (1)$$

where T stands for the set of all topics in microblogs and $\overline{DT'_u}$ denotes the mean interest degree of user u on these topics. r_ts are set as the probabilities of different topics' presence, which are computed according to the number of times unique words have been allocated to corresponding topics as expressed in matrix WT. In this case, the measure of interest similarities basically remains consistent with the topics' general influence. In the case of $sim(u, v) < 0$, we set $sim(u, v) = 0$ to avoid assigning a negative value to an edge of the network that is outside the scope of the research.

In this study, only the interest similarity between each friend and follower pair needs to be calculated, since the information can only spread from friend to follower, thus significantly reducing the time complexity and meeting the demands of practical application.

[3] http://nlp.stanford.edu/software/segmenter.shtml.

3.4. Interest interaction network construction

A directed network $D(V, E)$ is first constructed with the users and the friends/followers relationships among them. V denotes the vertex set, which contains all of the microblog users. E represents the edge set. Consistent with previous work (Page, Brin, Motwani et al., 1999), there is an edge between two users if there exists a "following" relationship between them and the arrow points from follower to friend.

The "surfer" randomly visits each user with a certain probability by following the corresponding edge in D, which is also regarded as the transmission probability of related information dissemination, since it reflects the influence of friend on follower. The probability matrix for information transmission, represented as P_t, is defined as follows.

Definition 2 (Transmission probability)**.** Each element of matrix P_t, i.e. the transmission probability from follower s_v to friend s_u of the random "surfer", is given by

$$P_t(u,v) = \frac{\log(|M_u|+1)}{\sum_{s_v \ follows \ s_w} \log(|M_w|+1)} * sim(u,v), \quad (2)$$

where $|M_u|$ denotes the number of microblogs released by s_u, and $sim(u, v)$ stands for the interest similarity between users s_u and s_v, with details as shown in Eq. (1).

This definition captures two intuitions. First is the assumption that user s_v follows many friends. Those friends release different numbers of microblogs, all of which will be directly visible to s_v. The more that friend s_u releases, the greater the portion of microblogs that s_v reads from s_u. Consequently, this brings about a higher influence on s_v, which results in a higher transmission probability from s_u to s_v. However, if user s_u publishes a large number of microblogs, it would create a subconscious boredom, which leads to the transmission probability being not linear with the number of microblogs. In this study, we adopt the logarithmic function to depict this relationship, as shown in the first term on the right-hand side (RHS) of Eq. (2).

Second, as implied by the homophily phenomenon discussed in Weng et al. (2010), s_u's influence on s_v is also related to the interest similarity between them; the greater the similarity, the greater the influence. Row-normalized matrix DT' is the result of topic distillation. Row DT'_u. represents the probability of user s_u's interest in different topics. The interest similarity between s_u and s_v can be calculated by the resemblance between the probability distributions, as illustrated in the second term on the RHS of Eq. (2).

3.5. Graph partitioning

Supposing that we are given the structure of an interest interaction network and there exist some isolated vertices and/or small groups of a few members, we then kick them out of the network since these communities formed by one or several users clearly cannot tell us anything of any worth.

Newman proposes a metric named *modularity* to evaluate the quality of graph partitioning (Newman, 2004, 2006). The modularity of a weighted network is defined as the sum of the weights of all of the edges included within subgraphs (after partitioning) subtracted by the expected edge weight sum under condition that edges were placed at random. A positive modularity implies a possible graph partitioning.

Leicht extends *modularity* to directed networks (Leicht & Newman, 2008). We can define user u_i's out-degree as $d_i^{(out)} = \sum_{j\in D} w_{ij}$ and in-degree as $d_i^{(in)} = \sum_{j\in D} w_{ji}$, where w_{ij} captures the weight of an edge directed from vertex i to j in network D. The sum of all of

the edge weights is defined as $m = \sum_{i \in D} d_i^{(out)}$ or $m = \sum_{i \in D} d_i^{(in)}$. The modularity of the partitioning is computed thusly

$$Q = \frac{1}{m} \sum_{ij} \left(w_{ij} - \frac{d_i^{(out)} d_j^{(in)}}{m} \right) \delta(C_i, C_j), \tag{3}$$

where $\delta(C_i, C_j)$ represents an impulse function. If vertices i and j are in the same community, i.e., $C_i = C_j$, then $\delta(C_i, C_j) = 1$; and $\delta(C_i, C_j) = 0$ otherwise.

Since the only contributions to the sum come from vertex pairs falling within the same cluster, we can combine these contributions and rewrite the sum over the clusters instead of the vertex pairs as

$$Q = \sum_{c=1}^{n_c} \left(\frac{l_c}{m} - \frac{d_c^{(out)} d_c^{(in)}}{m^2} \right). \tag{4}$$

Here, n_c is the number of clusters, l_c denotes the weight sum of edges connecting vertices within cluster c, $d_c^{(out)}$ stands for the sum of the out-degrees of the vertices in c, and $d_c^{(in)}$ represents the sum of the in-degrees.

In a network with extremely unbalanced communities, selecting an influential individual from each community would reduce the propagation effectiveness due to the extremely unbalanced sizes of the communities. To solve this problem, we introduce the *size-balanced increment of modularity* into the graph-partitioning process to balance community sizes as follows.

Definition 3 (Size-balanced increment of modularity)**.** The size-balanced increment of modularity obtained by dividing cluster c into two subgraphs is described as

$$Q_c = \frac{l_c}{m} - \frac{d_c^{(out)} d_c^{(in)}}{m^2}. \tag{5}$$

$$\Delta Q_c = Q_{c_1} + Q_{c_2} - Q_c. \tag{6}$$

$$\Delta Q'_c = \log s_c * \Delta Q_c, \tag{7}$$

where c_1 and c_2 represent the two sub-modules of c, and s_c denotes the number of individuals within c.

In a subgraph, the number of individuals shows an exponential increase with the path length of information spreading, which is exactly depicted using a logarithmic function in Eq. (7). For convenience, the number 10 is selected as the base of the logarithmic function.

The purpose here is to divide D into a pre-fixed number of subgraphs such that Q' is maximized. Leicht has proposed a very efficient and intuitive spectral-graph-theory-based approach to solve this optimization problem (Leicht & Newman, 2008). It first constructs a modularity matrix (D'') of the graph D, whose elements are described as

$$D'_{ij} = D_{ij} - \frac{d_i^{(out)} d_j^{(in)}}{m}. \tag{8}$$

$$D'' = D' + (D')^T, \tag{9}$$

where D_{ij} denotes an adjacency matrix element of graph D. Eigenanalysis is then executed on the symmetric matrix D'' to calculate its largest eigenvalue and the corresponding eigenvector ($\vec{v}$). Finally, D'' is divided into two subgraphs based on the plus-minus signs of the elements in $\vec{v}$.

To obtain the best possible modularity value, it is a common strategy in standard graph-partitioning issues to use spectral partitioning based on the graph Laplacian to gain an initial broad division of a network into two subnets, and then refine that partitioning by using the Kernighan–Lin approach. Each iteration of the fine-tuning algorithm in this study consists of the following steps: (i) Construct a list as a candidate vertex set by selecting among the vertices the ones that, when moved to the other subgraph, will give an increase in the modularity of the complete network. (ii) Repeatedly delete the vertex with largest modularity increase from the list and remove it to the other community. The process is executed until no further improvement in the modularity is possible. By building the candidate set, the computational complexity decreases significantly, since the search range is reduced considerably in each iteration.

It is important to note that it is incorrect, after first partitioning a network into two subgraphs, to simply delete the edges located between the two parts and then apply the approach again to each subgraph. This is because the degrees presenting in the definition, Eq. (3), of the modularity will decrease if edges are removed; thus, any subsequent maximization of modularity would maximize the wrong equation. Instead, the right way is to write the modularity matrix of a subgroup g with size s_g as

$$D''^{(g)}_{ij} = D''_{ij} - \delta_{ij} \sum_{k \in g} D''_{ik}|_{i \in g, j \in g}, \tag{10}$$

where δ_{ij} denotes the Kronecker δ-symbol, and $D''^{(g)}$ stands for the $s_g \times s_g$ matrix with elements retrieved by the labels i and j of vertices in group g.

The spectral approach is then recursively applied to each of the subgraphs to further partition them into smaller ones until the number of subgraphs satisfies the application requirement. In a real-world application, the number of influential individuals needed to be selected from the social media is always less than the actual number of communities in the social network; thus, no such situation exists in which the network is indivisible during the graph partitioning.

The main task of computation in this portion is to find the largest eigenvalues and the corresponding eigenvectors of the modularity matrices. This can be efficiently completed by the power iteration (Ipsen & Wills, 2006), the repeated multiplication of the matrix into a trial vector, which is able to scale up with the increase of the number of microblog users.

The partitioning process of the sparse symmetric modularity matrix D'' is elaborated in Algorithm 1.

3.6. Sorting by LeaderRank

In our proposed approach, size-balanced communities are first detected from the directed and weighted network, and we then select the most influential individual from each community by the modified LeaderRank algorithm.

LeaderRank is a random-walk-based ranking method (Lü et al., 2011). On the basis of PageRank, LeaderRank introduces a ground vertex g, which has two directed edges e_{gi} and e_{ig} connecting to every vertex i in the original network. For simplicity, the weights of e_{gi} and e_{ig} are set to the average weight of all of the edges in the original network. The rank score $r_j(t)$ of vertex j at discrete time t is given by (basing on a purely random walk process)

$$r_j(t) = \sum_{i=1}^{n+1} \frac{D_{ij}}{d_i^{(out)}} r_i(t-1). \tag{11}$$

Initially, $r_g(0) = 0$ for the ground vertex g, and $r_i(0) = 1$ for every other vertex i. At steady-state conditions, the rank score of the ground vertex is equally transferred to all of the other vertices to conserve scores without “following” edges. Therefore, the final score of vertex j is computed thusly

$$r_j = r_j(t_\infty) + \frac{r_g(t_\infty)}{n}. \tag{12}$$

Algorithm 1: Graph-partitioning process.

Input: D'', k; /*k is the pre-fixed number of influential individuals.*/

Output: CS; /*CS is the set of individual communities after graph partitioning.*/

1 calculate the community modularity of D'' as $Q_{D''}$;
2 split D'' into D''_1 and D''_2 by the spectral method with further fine-tuning, and then calculate the community modularity of D''_1 and D''_2 as $Q_{D''_1}$ and $Q_{D''_2}$, respectively.
3 size-balanced modularity increment $inc_{D''} = \log s_{D''} * (Q_{D''_1} + Q_{D''_2} - Q_{D''})$, map D'' (as the key) into $inc_{D''}$ (as the value) to build a new element of collection MC; /*MC is a mapping collection with each element containing a key and the corresponding value.*/
4 **while** $|MC| < k$ **do**
5 remove an element with the maximum value from MC, denote its key as DM;
6 divide DM into DM_1 and DM_2 by the spectral approach with fine-tuning strategy;
7 put DM_1 and DM_2 (with their size-balanced modularity increments) into MC as two new elements;
8 **end**
9 get all elements of MC, return their keys as set CS;

Here, $r_j(t_\infty)$ represents the rank score of vertex j in the stationary state.

Although LeaderRank is analogous to PageRank, it is more robust to attacks and more stable to noise than PageRank. More remarkably, LeaderRank is a parameter-free ranking algorithm.

3.7. Computation complexity

The most time-consuming process of our approach is the solution of the leading eigenvector of the modularity matrix. At first glance, it appears that the power iteration executes very slowly, taking $O(n^2)$ operations in each iteration because the modularity matrices are dense. However, we can subtly perform them much faster by taking full advantage of the particular structure of the matrix.

In the modularity matrix definition Eqs. (8) and (9), $D' = D - d^{(out)}(d^{(in)})^T/m$ and $D'' = D' + (D')^T$, where D denotes the adjacency matrix, and $d^{(out)}$ and $d^{(in)}$ are the out-degree and in-degree vectors. The product of D'' multiplying an arbitrary vector can be written as

$$D''x = (D + D^T)x - \left(\frac{d^{(out)}\left((d^{(in)})^T x\right)}{m} + \frac{d^{(in)}\left((d^{(out)})^T x\right)}{m} \right). \tag{13}$$

The first term on the RHS is a standard sparse matrix-vector multiplication taking time $O(m+n)$. The inner products $(d^{(in)})^T x$ and $(d^{(out)})^T x$ take time $O(n)$ to execute the vector-vector multiplication. Thus, the time taken to complete the multiplication in each round is $O(m+n)$, and generally $O(n)$ such multiplications are needed to obtain the leading eigenvector. Typically, the social network in which spammers have been eliminated is a sparse graph with $m \propto n$.

In conclusion, the overall running time of our proposed approach becomes $O(n^2)$.

According to the distributed computation (Sarma, Molla, Pandurangan et al., 2015) and the incremental computation (Bahmani, Chowdhury, & Goel, 2011) of PageRank, the iterative matrix-vector multiplication method can be performed distributively or incrementally for large-scale evolving networks. Sarma et al. (2015) provide a fast algorithm that takes $O(\sqrt{\log n}/\varepsilon)$ rounds in undirected graphs, where n represents the network size and ε denotes a fixed constant. Therefore, our approach can further accelerate the convergence if we implement the iterative matrix-vector multiplication procedure distributively or incrementally, which is planned future work.

4. Experimental evaluation

In this study, we compare our proposed approach with three other existing state-of-the-art methods, namely k-medoid, TwitterRank, and ClusterRank, respectively, on some synthetic networks generated by the Lancichinetti-Fortunato-Radicchi (LFR) model and a real-world microblogging network.

For ease of presentation, our proposed approach is annotated as PartitionRank throughout the comparisons in experiments.

4.1. Three compared methods

4.1.1. k-medoid

A novel approach is proposed by Zhang et al. (2013) to identify influential vertices in complex networks with community structure. The detailed process is as follows:

(i) An $n \times n$ information transfer probability matrix M on network $G = (V, E, W)$ is constructed, where n stands for the number of vertices in G, and element m_{ij} of M represents the information transfer probability through all paths from vertex i to j.
(ii) k medoids are then detected as k influential vertices by adopting the k-medoid clustering algorithm (Park & Jun, 2009) on M, which can be referred as a similarity matrix.

The time complexity is $O(n^3)$, where n denotes the number of vertices in the network.

4.1.2. TwitterRank

Weng et al. (2010) measure the influences of users in Twitter taking both the topic interest similarity between users and the following network into account.

The specific process of this method is as follows. First, topics that twitterers are interested in are extracted automatically by analyzing the content of their tweets. Second, a topic-specific social network among twitterers is constructed. Finally, the TwitterRank algorithm, an extension of PageRank, is applied to evaluate the influences of twitterers.

To keep analysis simple in this study, we construct the general influence topic-specific social network to perform the comparisons with our approach.

The computing time complexity is $O(n^2)$; here, n stands for the number of twitterers in the network.

4.1.3. ClusterRank

A local ranking algorithm called ClusterRank is proposed by Chen et al. (2013) that takes into account not only the number of followers and the followers' influences, but also the clustering coefficient.

Formally, the ClusterRank score s_i of vertex i is computed as

$$s_i = f(c_i) \sum_{j \in \Gamma_i} \left(d_j^{(in)} + 1 \right). \tag{14}$$

Here, Γ_i contains all of the followers of i, $d_j^{(in)}$ denotes the in-degree of vertex j, the term "+1" accounts for the contribution

of j itself, and the term $f(c_i)$ corresponds to the effect of i's local clustering. The local clustering plays a negative role in information spreading, so $f(c_i)$ is adopted as a decreasing function of clustering coefficient c_i, namely $f(c_i) = 10^{-c_i}$.

The time complexity is $O(m+n)$, where m and n represent the numbers of edges and vertices in the network, respectively.

4.2. Evaluation metrics

The influence maximization issue is tested on a directed and weighted network $G = (V, E, W)$. V and E are the sets of all of the individuals and links in the network, separately. n and m represent the numbers of items in V and E, respectively. W is the corresponding weight set of E, i.e., each link $(u, v) \in E$ from individual v to u corresponds to weight $w_{uv} \in W$.

4.2.1. Independent cascade model

To study the propagation process, we adopt a widely used information diffusion model, the independent cascade (IC) model (Kimura et al., 2010; Morid et al., 2014).

In the IC model, each link $(u, v) \in E$ is assigned a real value $\lambda_{uv} \in [0, 1]$ that is regarded as the probability of information dissemination through link (u, v), as illustrated in

$$\lambda_{uv} = 1 - (1 - \lambda)^{w_{uv}}, \tag{15}$$

where $\lambda \in (0, 1)$ denotes a specially designed propagation probability and w_{uv} captures the weight of link (u, v). Thus, the propagation probability λ_{uv} of each link (u, v) can be computed based on λ and w_{uv}.

In the IC model, some assumptions are brought forward: (1) The state of an individual is either active or inactive; an individual is active if he/she has taken the information. (2) Individuals can convert from being inactive to being active, but cannot convert from being active to being inactive. (3) Information spreading occurs only in discrete time steps $t \geq 0$. At time $t = 0$, the individuals in an initial set IA first become active, and all of the other individuals remain in the inactive state.

The propagation process of the IC model is formally described in Algorithm 2.

Algorithm 2: Independent cascade model.

Input: network $G = (V, E, W)$, λ and IA;
Output: AS; /*AS is the set of active individuals at the end of the propagation process.*/

```
AS = IA;
CA = IA; /*CA is the set of active individuals in the current time step.*/
NA = ∅; /*NA is the set of active individuals in the next time step.*/
while CA ≠ ∅ do
    foreach individual u ∈ CA do
        foreach individual v ∈ F(u) /*F(u) = {x|(u, x) ∈ E}.*/ do
            if v takes information from u with probability λ_uv
            and v ∉ AS, v ∉ NA then
                NA = NA ∪ v;
            end
        end
    end
    CA = NA; AS = AS ∪ NA; NA = ∅;
end
return AS;
```

4.2.2. Measures of information dissemination effects

In this paper, four measures of information diffusion effects are defined.

The first metric is the average number of active individuals (except initially active ones) at the end of the propagation process, denoted by $\sigma(A)$, which has been used in previous research (Kempe et al., 2003; Kitsak et al., 2010). $\sigma(A)$ is presented as

$$\sigma(A) = \frac{1}{n}\sum_{i=1}^{n} |AS_i(A)|, \tag{16}$$

where A represents a set of initially active individuals, $AS_i(A)$ indicates the output set of active individuals (excluding initially active ones) at the end of the ith propagation process of the IC model, $|AS_i(A)|$ denotes the number of individuals in $AS_i(A)$, and n captures the number of independent spreading simulations of the IC model.

The second criterion is the average scope of active individuals (not including initially active ones) at the end of multiple diffusion processes indicated as $\varphi(A)$, which is computed in

$$\varphi(A) = \frac{1}{H}\sum_{h=1}^{H}\sum_{v\in V} \omega(v, S_h(p, A)), \tag{17}$$

where H stands for the number of experimental groups, each group contains p independent spreading simulations of the IC model, V denotes the set of individuals in the network, and $S_h(p, A)$ represents the union of the output sets of active individuals (except initially active ones) in the p independent propagation processes of the hth experimental group. If $v \in S_h(p, A)$, then $\omega(v, S_h(p, A)) = 1$; otherwise, $\omega(v, S_h(p, A)) = 0$. With multiple spreading process simulations, increasing $\varphi(A)$ can ensure that the scope of information dissemination is more likely to be larger.

The third standard is the probability of an individual (not initially active) $v \in V$ taking the information, illustrated as

$$IP_v(A) = \frac{1}{q}\sum_{i=1}^{q} \omega(v, AS_i(A)), \tag{18}$$

where A is an initial set of active individuals, $AS_i(A)$ stands for the output set of active individuals (not containing initially active ones) at the end of the ith propagation process of the IC model, and q denotes the number of independent diffusing simulations. $\omega(v, AS_i(A)) = 1$ if $v \in AS_i(A)$, and $\omega(v, AS_i(A)) = 0$ otherwise.

The fourth measure is the amount of diffused words at the end of the spreading process denoted by $\psi(A)$, which is computed thusly

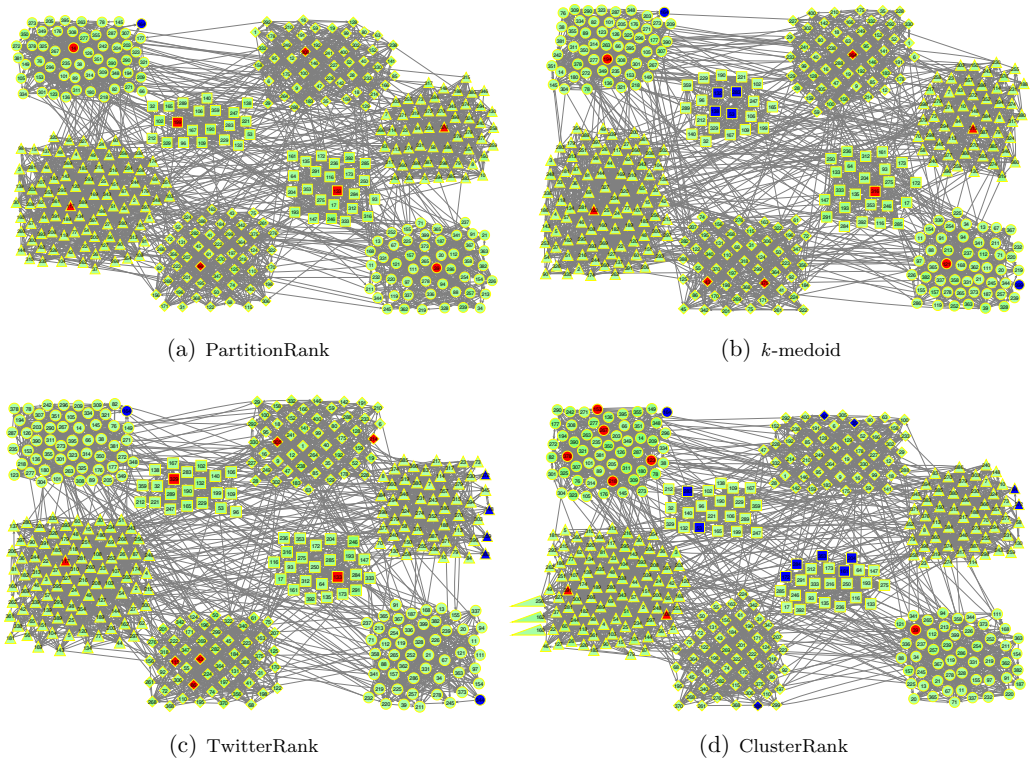
$$\psi(A) = \sum_{j=1}^{|T|}\sum_{v\in V} \omega(v, S_j(A)) * |T_j|, \tag{19}$$

where T stands for the set of short texts posted initially by an initial set of active individuals, $|T|$ represents the number of items in T, V is the set of individuals in the network, $|T_j|$ captures the number of words of the jth text in T, and $S_j(A)$ denotes the output set of active individuals who adopt the jth text. If $v \in S_j(A)$, then $\omega(v, S_j(A)) = 1$; otherwise, $\omega(v, S_j(A)) = 0$.

4.3. Experiments on artificial networks

4.3.1. LFR graphic generation model

In this study, we construct some artificial networks using the LFR model (Chen, Lü, Shang et al., 2012; Jiang, Perc, Wang et al., 2011) to evaluate the influential individual identification approaches. In the LFR model, both the individual degree and the community size follow the power-law distributions with exponents γ and η, separately. The number of individuals and the average degree are set to N and $\langle k \rangle$, respectively. The implementation steps of the LFR model are described in details as follows.



(a) PartitionRank (b) k-medoid

(c) TwitterRank (d) ClusterRank

Fig. 3. Effect comparison of four influential vertex identification methods on LFR-SU network. Built-in communities are represented by different shapes. Red marks detected influential vertices, green (from dark to light) denotes information adoption probabilities $IP_v(A)$ (Eq. (18)) (from high to low) of vertices, and blue indicates those in which $IP_v(A) = 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(1) Each individual is assigned a degree taken from a power-law distribution with exponent γ. The extreme degrees $k_{\min}$ and $k_{\max}$ are restricted so that the average degree equals $\langle k \rangle$. The configuration model (Molloy & Reed, 1998) is adopted to connect the individuals so as to keep the degree sequence.
(2) Each individual connects a fraction $1-\mu$ of its links with the other individuals of its community and a fraction μ with the rest of the individuals; μ is referred to as the mixing parameter.
(3) The sizes of the communities follow a power law with exponent η, so that all of the sizes sum up to the number N. The minimal and maximal community sizes $S_{\min}$ and $S_{\max}$ are selected so to keep the constraints $S_{\min} > k_{\min}$ and $S_{\max} > k_{\max}$, which guarantees that an individual of any degree can be included in at least one community.
(4) At first, all of the individuals are homeless, i.e., they do not belong to any community. In the first iteration, an individual is given to a randomly chosen community; if the number of its neighbors inside its community exceeds the community size, it remains homeless. In subsequent iterations, we assign a homeless individual to a randomly chosen community; if the latter is overloaded, we remove a randomly selected individual from the community and it becomes homeless. The procedure proceeds until there are no more homeless individuals.
(5) Several rewiring processes are conducted, to ensure the restriction on the fraction of internal neighbors expressed by the mixing parameter μ, such that the degrees of all of the individuals are kept the same, and only the split between internal and external degree is changed.
(6) Each link in the original network is doubled with two different-direction arrows assigned to the new links, to generate the corresponding directed network. For each individual, half of the links pointing to it are then randomly chosen and deleted from the network, such that the degrees of all of the individuals remain about the same, and there may exist a multi-dimensional relationship between any two individuals.
(7) In order to construct a weighted network, each link is assigned to a positive real number. Two parameters, α and μ_w, are needed in this process. The parameter α is adopted to assign a strength s_u to each individual u: $s_u = (k_u)^{\alpha}$. The parameter μ_w is used to allocate the internal strength: $s_u^{(in)} = (1-\mu_w)s_u$.

Lancichinetti et al. devised the corresponding software package[4] of the LFR model in the C++ programming language. In this study, many artificial networks are built according to the experimental requirements using this package.

4.3.2. Artificial networks construction

Three categories of artificial networks are generated through the LFR model to compare our PartitionRank approach with the other three methods.

(1) LFR-SU
A small unweighted (SU) and directed network. The number of vertices is 400; the average degree is 15; the maximum degree is 20; the degree distribution exponent is −1.8; the community sizes are between 20 and 60; the community size distribution exponent is −1.2; the mixing parameter is 0.1; and the number of communities is 8.
(2) LFR-LU
A large unweighted (LU) and directed network. The number of vertices is 2000; the average degree is 30; the maximum

[4] http://santo.fortunato.googlepages.com/benchmark.tgz.

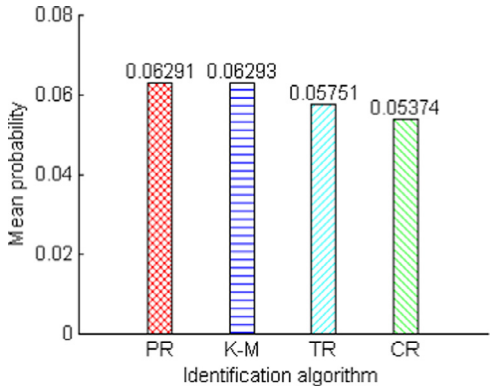


Fig. 4. Mean adopting probability of all of the vertices for four influential vertex identification approaches on LFR-SU network. Here, PartitionRank, *k*-medoid, TwitterRank and ClusterRank are abbreviated as PR, K-M, TR and CR, respectively.

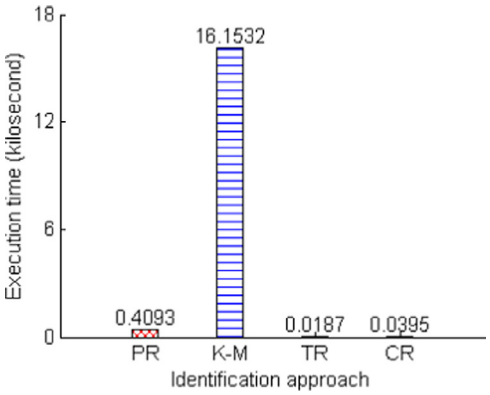


Fig. 5. Execution time of different influential vertex identification algorithms on LFR-SU network. Here, we separately abbreviate PartitionRank, *k*-medoid, TwitterRank and ClusterRank by PR, K-M, TR and CR.

degree is 40; the degree distribution exponent is −1.8; the community sizes are between 40 and 100; the community size distribution exponent is −1.2; the mixing parameters are 0.1, 0.3 and 0.5, respectively; and the number of communities is 28.

(3) LFR-LW

A large weighted (LW) and directed network. The number of vertices is 2000; the average degree is 30; the maximum degree is 40; the degree distribution exponent is −1.8; the community sizes are between 40 and 100; the community size distribution exponent is −1.2; the strength assignment parameter is 0.25; the internal strength parameters are 0.3 and 0.5 separately; and the number of communities is 28.

4.3.3. Results on artificial networks

In this study, when the average number of active individuals $\sigma(A)$ (Eq. (16)) and the adopting probability $IP_v(A)$ (Eq. (18)) are calculated, the number of simulations of information dissemination is 100,000 for all of the experiments. When the average scope of active individuals $\varphi(A)$ (Eq. (17)) is calculated, 100 groups of experiments are implemented and each group consists of 1000 independent spreading simulations.

For the LFR-SU network, we set $k = 8$ according to the actual number of communities. The TwitterRank and ClusterRank methods choose influential vertices only from five and three of eight communities separately, such that some vertices in the communities with no influential vertices can only obtain the information with a probability of approximately zero under condition $\lambda = 0.1$ (Fig. 3(c) and 3(d)). The PartitionRank and *k*-medoid approaches, however, almost detect an influential vertex from each community (Fig. 3(a) and 3(b)). With an initially active vertex located in the vital position of each module, the remaining members would maximize the probability for acquiring information, which leads to the highest performance of the PartitionRank algorithm on valuation standard $IP_v(A)$ (Eq. (18)). Exceptionally, the adopting probability of the vertex labeled with number 164 always equals 0 for all identification algorithms since $d_{164}^{(out)} = 0$.

Fig. 4 illustrates the mean adopting probability of all of the vertices for different identification methods on the LFR-SU network, from which we can conclude that the PartitionRank and *k*-medoid schemes are vastly superior to the other two approaches.

As for the time-consumption aspect, the *k*-medoid algorithm runs particularly slowly compared with the other three methods on the LFR-SU network (Fig. 5).

In the LFR-LU networks, k is initially set equal to 8, and then gradually increases with a step size of 10 until reaching 28. Fig. 6 compares the information spreading effects of these methods on the LFR-LU networks by the average number of active vertices $\sigma(A)$ (Eq. (16)) and the average active scope $\varphi(A)$ (Eq. (17)) at different mixing parameters μs (stated in Section 4.3.1) and propagation probabilities λs (Eq. (15)). For all of the different μs and λs, the PartitionRank algorithm gains the highest values of $\sigma(A)$ and $\varphi(A)$ despite the fact that it only has a small advantage over the *k*-medoid method. However, as the μ and λ values increase, which means that the community structure grows ever more obscure, the differences become small.

Table 1 demonstrates the execution efficiencies of these four algorithms on the LFR-LU networks, where the *k*-medoid scheme consumes the most time, as expected. It is worth noting that the time consumption of running the LDA procedure is deducted, since the synthetic networks are generated without performing the topic distillation in this study. Nevertheless, upon comparison with the overall execution time complexity of $O(n^2)$, the required time to perform the LDA process with the complexity of $O(n)$ can be ignored in the PartitionRank and TwitterRank approaches, where n denotes the number of vertices in the network.

In Fig. 7, we assess the performance of the four algorithms employed on the LFR-LW networks at different internal strength parameters μ_ws (see Section 4.3.1) and propagation probabilities λs (Eq. (15)). For all of the different μ_ws and λs, the $\sigma(A)$ and $\varphi(A)$ values of the PartitionRank and *k*-medoid algorithms are still higher than those of the TwitterRank and ClusterRank methods, wherein the scheme proposed in this study achieves the highest performance. However, as the μ_w and λ values increase, the performance gap becomes small in the same way.

As displayed in Table 2, the *k*-medoid approach remains the most time-consuming of all of the compared methods on the LFR-LW networks.

Experimental results on artificial networks with various settings demonstrate that our algorithm almost always produces the best solution in comparable time among state-of-the-art methods, which means that the PartitionRank approach can satisfy the application requirements of large-scale expert and intelligent systems under the independent cascade model.

4.4. Experiments on real-world network

4.4.1. Network construction of microblogging

Tencent microblogging, similar to Twitter, adopts a social-networking model named “following”, in which each user can “follow” anyone from whom he/she wants to receive microblogs without permission. Through the end of 2012, the number of registered users on the Tencent microblogging platform has exceeded

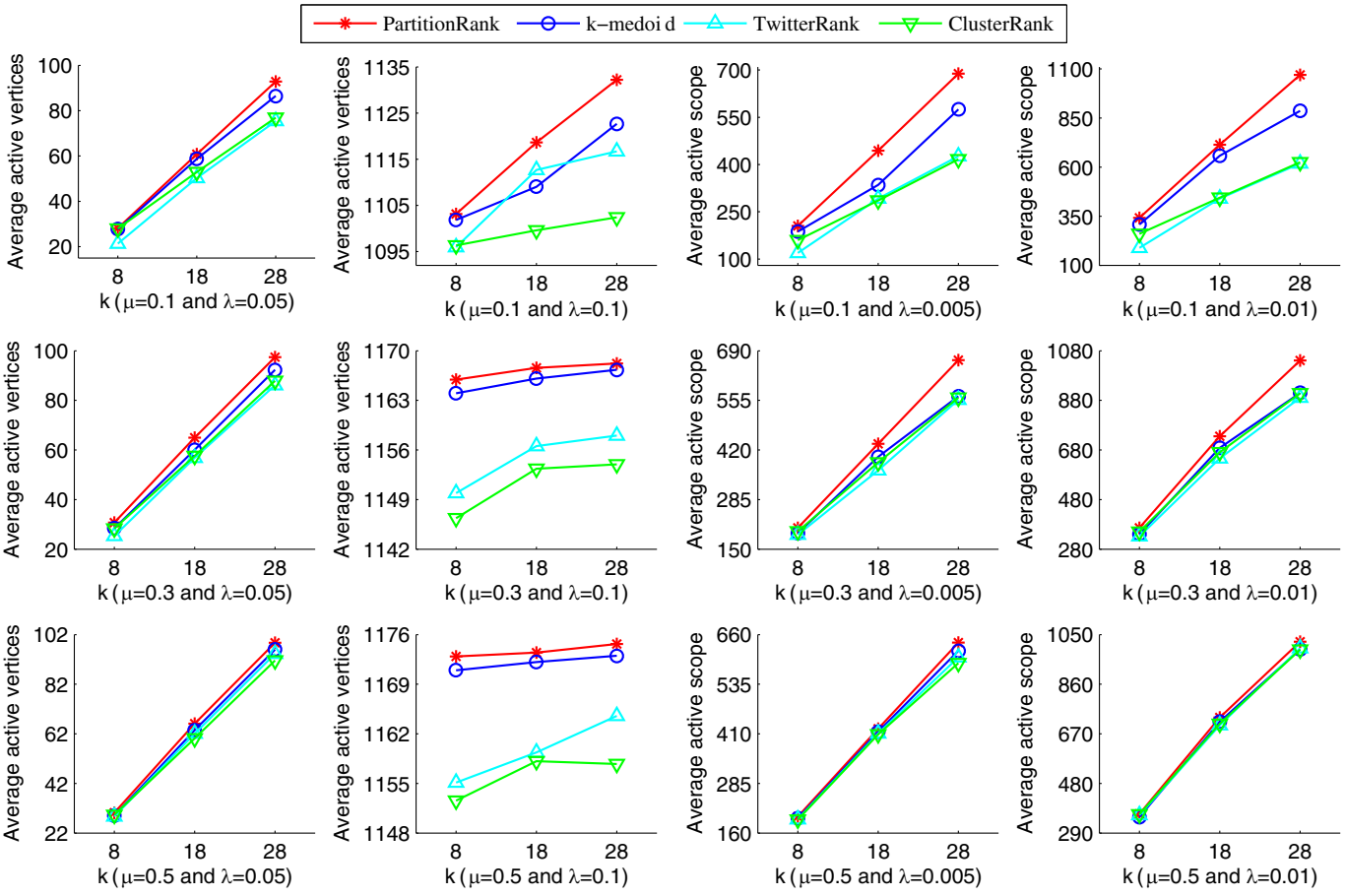


Fig. 6. Effect comparison of four influential vertex identification approaches on LFR-LU networks.

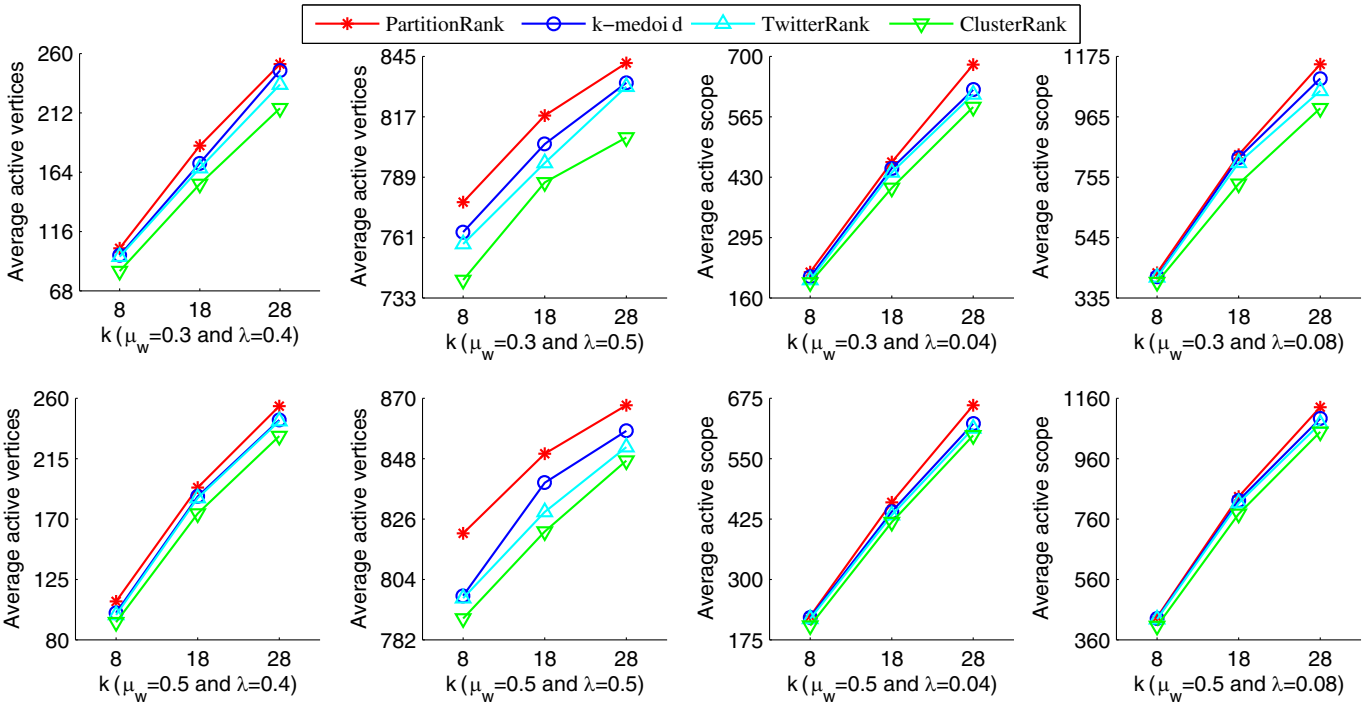


Fig. 7. Effect comparison of four influential vertex identification algorithms on LFR-LW networks.

Table 1
Execution time of different influential vertex identification methods on LFR-LU networks (not including LDA process time consumption).

Parameters			Time (s)			
μ	λ	k	PartitionRank	k-medoid	TwitterRank	ClusterRank
0.1	0.005	8	13,487	55,025	968	227
		18	14,595	55,058	975	239
		28	14,663	56,909	982	257
	0.01	8	13,487	81,459	968	227
		18	14,595	82,228	975	239
		28	14,663	82,696	982	257
	0.05	8	13,487	832,025	968	227
		18	14,595	866,543	975	239
		28	14,663	893,566	982	257
	0.1	8	13,487	208,978,733	968	227
		18	14,595	212,110,411	975	239
		28	14,663	227,410,462	982	257
0.3	0.005	8	13,569	55,910	819	230
		18	15,396	56,197	832	242
		28	15,310	57,039	841	259
	0.01	8	13,569	81,876	819	230
		18	15,396	82,610	832	242
		28	15,310	82,884	841	259
	0.05	8	13,569	854,872	819	230
		18	15,396	879,032	832	242
		28	15,310	896,964	841	259
	0.1	8	13,569	214,307,601	819	230
		18	15,396	219,783,575	832	242
		28	15,310	229,133,579	841	259
0.5	0.005	8	15,146	56,714	677	247
		18	16,569	56,908	681	258
		28	16,687	57,225	691	265
	0.01	8	15,146	82,105	677	247
		18	16,569	82,683	681	258
		28	16,687	82,942	691	265
	0.05	8	15,146	862,299	677	247
		18	16,569	892,087	681	258
		28	16,687	907,808	691	265
	0.1	8	15,146	219,854,799	677	247
		18	16,569	223,206,054	681	258
		28	16,687	238,730,887	691	265

Table 2
Execution time of different influential vertex identification approaches on LFR-LW networks (not including LDA process time consumption).

Parameters			Time (s)			
μ_w	λ	k	PartitionRank	k-medoid	TwitterRank	ClusterRank
0.3	0.04	8	17,576	52,886	1001	260
		18	18,748	53,490	1013	271
		28	18,885	54,172	1050	281
	0.08	8	17,576	83,997	1001	260
		18	18,748	85,038	1013	271
		28	18,885	85,904	1050	281
	0.4	8	17,576	2,073,651	1001	260
		18	18,748	2,136,160	1013	271
		28	18,885	2,152,149	1050	281
	0.5	8	17,576	94,758,742	1001	260
		18	18,748	97,147,645	1013	271
		28	18,885	97,673,281	1050	281
0.5	0.04	8	21,661	55,547	665	271
		18	22,350	56,012	697	280
		28	22,565	57,203	707	290
	0.08	8	21,661	85,157	665	271
		18	22,350	86,271	697	280
		28	22,565	87,068	707	290
	0.4	8	21,661	2,145,169	665	271
		18	22,350	2,239,384	697	280
		28	22,565	2,276,103	707	290
	0.5	8	21,661	101,331,382	665	271
		18	22,350	101,972,710	697	280
		28	22,565	102,230,236	707	290

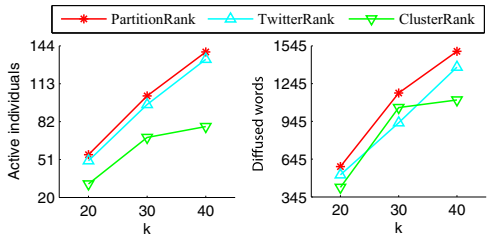


Fig. 8. Effect comparison of three influential individual identification algorithms on real microblogging network.

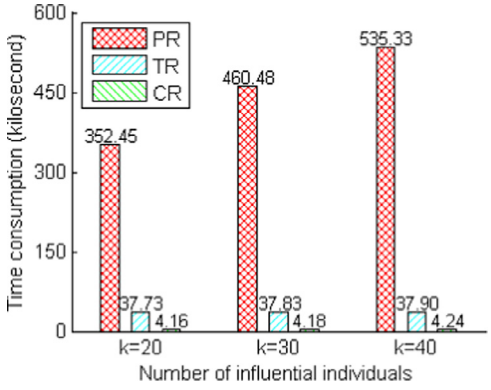


Fig. 9. Execution time of different influential individual identification approaches on real microblogging network. Here, PR, TR and CR are abbreviations of PartitionRank, TwitterRank and ClusterRank, respectively.

540 million. In the meantime, the number of daily active users has reached up to 100 million. Furthermore, Tencent microblogging has provided a uniform free application program interface (API[5]) for developers to acquire the full user data. Given the advantages above, Tencent microblogging is employed as the experimental platform for the performance assessment on real-world social media in this study.

On average, there are dozens of friends on the entire site for each user. If we stochastically select some users to build the "following" network, then it cannot accurately depict the social relationships of these users. Therefore, we begin with a specified user who is identified as the initial individual of the "following" network, and then all of the followers of this user are recruited into the network. After that, we add all of the followers of these newcomers in the same way, until the number of individuals in the network satisfies the experimental requirement. Meanwhile, those users having more than 1000 friends or followers are eliminated, since we regard them as friendship abusers or stars.

We perform the experiments with two batches of microblog data related to the same 12,746 users, which are collected in June and August 2015 for influential individuals identification and performance verification, respectively. There exist 83,809 "following" relationships among these users in June.

The LDA model is conditioned on three parameters, i.e., topic number T and Dirichlet hyper-parameters α and β. In this study, they are set as $T = 100$, $\alpha = 50/T$, and $\beta = 0.1$. The different choice of these parameters has implications for the results of the topic distillation. Nevertheless, it is not investigated since the focus of this study is how to identify the influential individuals in microblogging networks. Indeed, the results and perspective of this study are not limited by the very specific values of these three parameters.

4.4.2. Results on microblogging platform

For the experiments on a real-world network, the numbers of identified influential individuals ks are set to 20, 30 and 40 separately. In this case, the k-medoid approach is dismissed since the required memory exceeds the maximum allowed by our system, which is 16 GB.

Fig. 8 compares the information spreading effects of the remaining three compared methods on the microblogging network by the active vertex number $\sigma(A)$ (Eq. (16)) and the amount of diffused words $\psi(A)$ (Eq. (19)). From the results of all of the different ks, one can draw a conclusion that our scheme outperforms the other methods in $\sigma(A)$ values; in particular, our approach gains the highest $\psi(A)$ values with significant superiority.

Similarly, Fig. 9 shows the execution efficiencies of these related algorithms. Note that our proposed approach can satisfy the needs of practical applications, although it is significantly slower than the other two methods.

[5] http://dev.t.qq.com.

According to the experimental results on the microblogging site, which are independent of any simulation model, we can conclude that the PartitionRank approach significantly outperforms the other compared methods within acceptable time. In other words, our scheme illustrates broad prospects for intelligent applications in social media similar to microblogging, including opinion propagation, guidance of political movements, and acceptance of technologies.

4.5. Experimental results discussion

Consistent with the experimental results above, we further carry out a deep analysis and discussion of the results of comparing our approach to the other three methods.

For the k-medoid method, the transmission probability and spreading mechanism that can impact the spreading effects are taken into account, but the computation cost is particularly high, so that it cannot adapt to large-scale social networks. In contrast, our approach first partitions the propagation probability network into a pre-fixed number of size-balanced communities, and then identifies the most influential vertex from each community as the result set, which obviously can reduce the time complexity. Meanwhile, the spreading mechanism and propagation probability are implicitly included in the graph-partitioning process. In addition, our method even accounts for the interest similarity among users of social media, which further improves its performance in influential individuals identification.

The TwitterRank method is a PageRank-like algorithm, in which the interest similarities on different topics between each pair of friend and follower are considered to measure the topic-sensitive influences of twitterers. However, the identified influential individuals may be distributed in the communities with large sizes, and can only influence vertices of the same communities in a short time span when the propagation probability is small. Through graph partitioning, our algorithm can select influential individuals from different communities in a balanced way. Thus, the influential individuals detected by the PartitionRank approach can diffuse more information than the TwitterRank method.

The ClusterRank method emphasizes the negative effects of the local clustering on spreading dynamics, and fails to separate the influential individuals by a distance to avoid repeatedly influencing the same vertices. On contrary, in addition to the interest similarity calculation, our method identifies only the most influential individual from each size-balanced community, which can reduce or even eliminate the overlapping effects to achieve a wider range of information dissemination.

5. Conclusions and future work

In this study, we propose an overall approach named PartitionRank for influential individual identification in microblogging networks with some advantages. On the one hand, in accordance with the "homophily" phenomenon, the topical similarities among users complement the interaction relationships to construct the interest interaction network that can accurately simulate the information propagation probability. On the other hand, the proposed scheme detects the most influential individual from each of a prefixed number of communities in a balanced way, which can influence many more users of the social networks. Differing from ordinary community detection methods, we introduce the size balance model into the graph partitioning, which further improves the spreading effectiveness.

The current design of our algorithm only accounts for the number of microblogs a user releases in the transmission probability estimation process. In the future, we intend to improve this by incorporating more interactions between users, e.g., mention/reply. In addition, the computation cost of our approach mainly depends on the graph-partitioning part; therefore, developing a top-down community detection method with low time complexity is another object of future research. Last but not least, we plan to further validate the performance of the proposed approach in large-scale microblogging datasets.

Acknowledgments

This study is partially funded by the National Key Research and Development Program of China (Grant No. 2017YFC0907505), the National Natural Science Foundation of China (Grant Nos. 61772128 and 61303096), the Fundamental Research Funds for the Central Universities (Grant No. 16D111208), the Shanghai Natural Science Foundation (Grant No. 17ZR1400200), and the Xinjiang Social Science Foundation (Grant No. 2015BGL100).

References

Bahmani, B., Chowdhury, A., & Goel, A. (2011). Fast incremental and personalized PageRank. In *Proceedings of the thirty-seventh international conference on very large data bases: 4* (pp. 173–184).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of machine Learning research, 3*, 993–1022.

Bonacich, P. (2007). Some unique properties of eigenvector centrality. *Social Networks, 29*(4), 555–564.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems, 30*(1), 107–117.

Chen, D., Lü, L., Shang, M. S., et al. (2012). Identifying influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications, 391*(4), 1777–1787.

Chen, D. B., Gao, H., Lü, L., et al. (2013). Identifying influential nodes in large-scale directed networks: The role of clustering. *PLoS ONE, 8*(10), e77455.

Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social networks, 1*(3), 215–239.

Heinrich, G. (2008). Parameter estimation for text analysis. *Technical Report*. University of Leipzig.

Ipsen, I. C. F., & Wills, R. S. (2006). Mathematical properties and analysis of Google's PageRank. *Boletín de la Sociedad Espaòla de Matemática Aplicada, 34*, 191–196.

Jiang, L. L., Perc, M., Wang, W. X., et al. (2011). Impact of link deletions on public cooperation in scale-free networks. *EPL (Europhysics Letters), 93*(4), 40001.

Kaiser, C., Schlick, S., & Bodendorf, F. (2011). Warning system for online market research – identifying critical situations in online opinion formation. *Knowledge-Based Systems, 24*(6), 824–836.

Keeling, M. J., & Rohani, P. (2008). *Modeling infectious diseases in humans and animals*. Princeton University Press.

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 137–146).

Kim, Y. A., & Song, H. S. (2011). Strategies for predicting local trust based on trust propagation in social networks. *Knowledge-Based Systems, 24*(8), 1360–1371.

Kimura, M., Saito, K., Nakano, R., et al. (2010). Extracting influential nodes on a social network for information diffusion. *Data Mining and Knowledge Discovery, 20*(1), 70–97.

Kitsak, M., Gallos, L. K., Havlin, S., et al. (2010). Identification of influential spreaders in complex networks. *Nature Physics, 6*(11), 888–893.

Leicht, E. A., & Newman, M. E. J. (2008). Community structure in directed networks. *Physical Review Letters, 100*(11), 118703.

Leskovec, J., Krause, A., Guestrin, C., et al. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the thirteenth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 420–429).

Li, Q., Zhou, T., Lü, L., et al. (2014). Identifying influential spreaders by weighted LeaderRank. *Physica A: Statistical Mechanics and its Applications, 404*, 47–55.

Lü, L., Zhang, Y. C., Yeung, C. H., et al. (2011). Leaders in social networks, the delicious case. *PLoS ONE, 6*(6), e21202.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology, 27*, 415–444.

Molloy, M., & Reed, B. (1998). The size of the giant component of a random graph with a given degree sequence. *Combinatorics, probability and computing, 7*(03), 295–305.

Morid, M. A., Shajari, M., & Hashemi, A. R. (2014). Defending recommender systems by influence analysis. *Information Retrieval, 17*(2), 137–152.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E, 69*(6), 066133.

Newman, M. E. J. (2006). Modularity and community structure in networks. In *Proceedings of the national academy of sciences: 103* (pp. 8577–8582).

Nguyen, N. P., Dinh, T. N., Xuan, Y., et al. (2011). Adaptive algorithms for detecting community structure in dynamic social networks. In *Proceedings of the thirty–firth IEEE international conference on computer communications* (pp. 2282–2290).

Ohsaka, N., Akiba, T., Yoshida, Y., et al. (2014). Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In *Proceedings of the twenty-eight AAAI conference on artificial intelligence* (pp. 138–144).

O'Mahony, M. P., & Smyth, B. (2010). A classification-based review recommender. *Knowledge-Based Systems, 23*(4), 323–329.

Page, L., Brin, S., Motwani, R., et al. (1999). The PageRank citation ranking: Bringing order to the web. *Technical report*. Stanford InfoLab.

Palla, G., Pollner, P., Barabási, A. L., et al. (2009). *Social group dynamics in networks*. Berlin Heidelberg: Springer. Adaptive Networks:11-38

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications, 36*(2), 3336–3341.

Pei, S., Muchnik, L., Andrade, J. S., et al. (2014). Searching for superspreaders of information in real-world social media. *Scientific Reports, 4*, 5547.

Rahimkhani, K., Aleahmad, A., Rahgozar, M., et al. (2015). A fast algorithm for finding most influential people based on the linear threshold model. *Expert Systems with Applications, 42*(3), 1353–1361.

Rogers, E. M. (1995). *Diffusion of innovations* (4th edition). New York: Free Press.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika, 31*(4), 581–603.

Sarma, A. D., Molla, A. R., Pandurangan, G., et al. (2015). Fast distributed PageRank computation. *Theoretical Computer Science, 561*, 113–121.

Weng, J., Lim, E. P., Jiang, J., et al. (2010). TwitterRank: Finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on web search and data mining* (pp. 261–270).

Zamora-López, G., Zhou, C., & Kurths, J. (2010). Cortical hubs form a module for multisensory integration on top of the hierarchy of cortical networks. *Frontiers in Neuroinformatics, 4*(1), 27–39.

Zhang, X., Zhu, J., Wang, Q., et al. (2013). Identifying influential nodes in complex networks with community structure. *Knowledge-Based Systems, 42*, 74–84.