Contents lists available at ScienceDirect

Information Systems

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

Community-diversified influence maximization in social networks

Jianxin Li [a,*], Taotao Cai [a,*], Ke Deng [b], Xinjue Wang [b], Timos Sellis [c], Feng Xia [d]

[a] School of Information Technology, Deakin University, Geelong, Australia
[b] Department of Computer Science & Informaton Technology, Royal Melbourne Institute of Technology, Melbourne, Australia
[c] Data Science Research Institute, Swinburne University of Technology, Hawthorn, Australia
[d] School of Science, Engineering and Information Technology, Federation University Australia, Ballarat, Australia

ARTICLE INFO

Article history:
Received 23 October 2018
Received in revised form 29 July 2019
Accepted 22 March 2020
Available online 26 March 2020
Recommended by Dennis Shasha

Keywords:
Social community
Influence maximization
Diversified influence propagation

ABSTRACT

To meet the requirement of social influence analytics in various applications, the problem of influence maximization has been studied in recent years. The aim is to find a limited number of nodes (i.e., users) which can activate (i.e. influence) the maximum number of nodes in social networks. However, the community diversity of influenced users is largely ignored even though it has unique value in practice. For example, the higher community diversity reduces the risk of marketing campaigns as you should not put all your eggs in one basket; the diversity can also prolong the effect of a marketing campaign in the future promotion. Motivated by this observation, this paper investigates *Community-diversified Influence Maximization* (CDIM) problem to efficiently find k nodes such that, if a message is initiated and spread by the k nodes, the number as well as the community diversity of the activated nodes will be maximized at the end of propagation process. This work proposes a metric to measure the community-diversified influence and addresses a series of computational challenges. Two algorithms and an innovative *CPSP-Tree* index have been developed. This study also investigates the situation that community definition is not specified. The effectiveness and efficiency of the proposed solutions have been verified through extensive experimental studies on five real-world social network datasets.

1. Introduction

Social media has become an essential platform for organizations to broadcast their policies, for companies to advertise their products, and for people to propagate their opinions. This stimulates the study of *influence maximization* (IM) problem. The intuition is to select k influential nodes (a node represents a social media user) in social networks, known as *seeds*, convince them to adopt a product (or a service, an idea, a political opinion, etc.), and utilize the "word-of-mouth" effect to spread the information with attempt to activate other nodes in the social networks to adopt it as well. The IM problem is to decide which k nodes in the social networks should be selected such that the number of nodes activated (or influenced) in the social networks are maximized [1,2]. A large body of recent works have studied the IM problem with additional considerations. *Topic-aware* IM problem considers the topics of information to be spread. The possibility that a node adopts the information is affected by the interest of the node to the topic [3–5]. *Competitor-aware* IM problem models the propagation rate of information over social networks in order to activate more nodes before they are influenced by information from competitors [6–8]. *Location-aware* IM problem contemplates the physical locations of nodes to be influenced [9,10].

The studies of IM problem assume the information propagation initiated from seeds follows some diffusion model. The Independent Cascade (IC) diffusion model is widely accepted and studied [2]. Under IC diffusion model, the influence of a seed decays continuously when the information spread from one node to the next in social networks until the influence is too trivial to be noticed. At the end of propagation process, the nodes are activated if it has sufficiently high probability to be influenced.

While the focus of existing studies is on the maximum number of nodes to be activated, the community diversity of the activated nodes is largely ignored. The higher community diversity means the activated nodes are from more communities which has critical value in practice. In real world marketing, the diversity of target audience could bring many benefits [11]. As you should not put all your eggs in one basket, the diversity could reduce the risk of marketing campaigns. Also, the diversity can prolong the effect of a marketing campaign. For example, a fraction of activated nodes becomes the registered users of a product; even though it is hard to estimate who will become the registered users and even the number of registered users, it is reasonable to assume this happens randomly among the activated users; the following promotions of the same or similar products in the future will be

[*] Correspondence to: 1 Gheringhap St, Geelong, VIC, Australia.
E-mail addresses: jianxin.li@deakin.edu.au (J. Li), taotao.cai@deakin.edu.au (T. Cai).

https://doi.org/10.1016/j.is.2020.101522

able to reach these registered users directly, in other words, more communities.

This observation motivates us to investigate Community-diversified Information Maximization (CDIM) problem. The aim is to select k nodes in a social network such that the number of activated nodes and the community diversity of the activated nodes can be maximized concurrently if the selected k nodes spread a message following IC diffusion model. This work is the first effort to investigate the community-diversified influence maximization problem. The contributions are summarized:

- This work introduces the community-diversified influence maximization problem which has unique values in market campaigns. Due to the community-diversity, it can reduce the risk of market campaigns. Also, it breaks through the limit of existing studies which assume the activated nodes are the dead-end of information propagation. In many applications in market campaigns, however, it is reasonable that the activated nodes can be explored in the future to spread other information.
- A deliberately designed metric has been proposed to evaluate CDIM by considering both the number of activated nodes and the diversity of their influenced communities concurrently.
- This study has developed two algorithms to solve CDIM problem efficiently, i.e., greedy and upper bound based algorithms. The greedy algorithm comes with reasonable approximation bound. To enable more efficient processing, the upper bound of community-diversified influence has been explored to minimize the search space of exploring seed candidates. As a further step to accelerate the efficiency, the *Community-aware Partial Shortest Path tree* (CPSP-Tree) has been designed to estimate the influence of candidate nodes in social networks.
- Extensive experimental evaluation have been conducted on five real-world datasets. The test results have demonstrated the superiority of CDIM solutions in terms of effectiveness and the processing efficiency.

The remainder of this paper is organized as follows. The related work is presented in Section 2. Section 3 defines the CDIM problem and the objective function. The monotonous and submodular properties of the problem has been proved in Section 3.3. In Section 4, we propose two algorithms to solve the CDIM problem. Section 5 introduces CPSP-Tree and the associated processing method. Then, we discuss the solution in Section 6 if the community detection method is not available. We analyze and discuss the experimental results in Section 7 and conclude this paper in Section 8.

2. Related work

2.1. Influence maximization

Kempe et al. [2] has proposed two discrete influence propagation models, *Independent Cascade* (IC) model and *Linear Thresholds* (LT) model. Based on the two models, there are lots of work focusing on influence maximization problem, e.g., [1,5,12–15]. The aim is to select a limited number of nodes in social networks as *seeds* such that, the information initiated by the seeds in the social networks will activate the maximum number of nodes at the end of influence propagation process following either IC model or LT model. While considering the number, the existing studies ignore the community diversity of activated nodes in the social networks.

Compared to other studies, the problem defined in [11] is more relevant to CDIM. The solution in [11] aims to enforce the diversity on seeds. The idea behind is that if the seeds are diverse, then the resultant activated nodes would be diverse too. In [11], the diversity is defined on categories, which is similar to Topic-aware IM [5]. Given a set of node, the higher diversity means more nodes belong to more categories. Different from [11], the diversity concerns in CDIM is the number of different communities which are featured by dense internal connectivity and loose external connectivity in social networks. Also, in the context of community diversity, it is unclear whether the diversity of seeds is related to the optimal solution of CDIM or not due to the complexity of the problem.

The personalized social influential tags exploration problem has been studied [16]. However, it is irrelevant to CDIM problem. Given a target user, from a set of tags which characterize the content propagated in a social network, it aims to exact k tags that can maximize the user's social influence. In [17], the community-based greedy algorithm has been studied to mine a set of top-k influential nodes in a given mobile social network such that the number of activated nodes is maximized using an extended IC model. The greedy algorithm is expensive for solving the influence maximization problem on a large-scale network. So it proposes a community based greedy algorithm which mine the influential nodes in each community rather than the whole network. Other studies related to social network influence include the most influential community search in a social network [18] and searching objects with high influence in terms of spatial closeness [19,20].

2.2. Community detection in social networks

A great deal of work has been devoted to find communities in large networks, and much of this has been devoted to formalize the intuition that a community is a set of nodes that has more and/or better links between its members than with the remainder of the network.

A line of work is to discover communities based on explicit community model like k-core [21] and k-truss [22]. The k-core of a graph is the largest subgraph within which each node has at least k connections. In the induced subgraph (i.e., a k-core community), since it only requires each node has k neighbors, two nodes may have large hops (i.e., less cohesive). Given a graph G, the k-truss of G is the largest subgraph in which every edge is contained in at least $(k-2)$ triangles within the subgraph. The k-truss is a type of cohesive subgraph defined based on triangle which models the stable relationship among three nodes. With edge connectivity constraints, the induced subgraph (i.e., a k-truss community) is connected and cohesive. But the connectivity is so strong that k-truss can only be used to discover communities of very small size (i.e., not a society). In [23], the explicit community model called k-r Maximal Cliques (krMC) considers the social network influence of each community.

The other line of research focuses on implicit community detection models which concerns the global connectivity of the social network and the discovered communities often have small cohesiveness. Newman and Girvan in [24] proposed a quantitative measure, called modularity, to assess the quality of community structures, and formulated community discovery as an optimization problem. The key idea is similar to graph partitioning, which iteratively removes the edge with the highest betweenness score. Betweenness based community detection metric was also studied by Girvan and Newman in [25]. Ruan and Zhang [26] proposed a more efficient spectral algorithm to find high quality communities by applying k-way partitioning and recursive 2-way partitioning strategies [27]. Satuluri and Parthasarathy in [28] developed efficient Markov clustering algorithms to identify communities by using stochastic flow technique. The key idea in it is to enhance flow to well-connected nodes, i.e., rich get richer and poor get poorer. LPA [29] has been evaluated and recommended to be the better choice as an accurate and efficient community detection technique in the recent studies [30] as well as [31]. It works as follows. Each node in a social network is first given a unique label. At every iteration, each node is updated by choosing the label which most of its neighbors have. If a node happens to be multiple labels, then one of these would be selected randomly. After several iterations, the communities will be uncovered via the labels where each label represents a community.

Moreover, other community detection methods have been studied with edge content consideration in [32], with clique definition and parallel algorithm in [33]. More details can be found in surveys [30,34].

2.3. Diversity

The diversity of search results has attracted the attention from researchers in different fields. For instance, the diversified keyword search have been studied in database community [35–37], in Web search [38,39], in information retrieval [40,41]. In these studies, a max-sum type objective function is typically used to concurrently consider both relevance and diversity of the search results. The interested readers are referred to read the survey [42]. The problem in all the above works are very different from our CDIM problem. Given search criteria, their aim is to find items which are most relevant and diverse. In contrast, the objective of CDIM problem is to find seeds which can influence maximum number of nodes from as many different communities as possible.

3. Preliminaries and problem definition

3.1. Preliminary

A social network is modeled as a directed graph $G = (V, E)$, where a node in V represents one social media user, the edge (u, v) in E represents the (*follower*, *followee*) relationship, and each edge (u, v) is associated with a value $w_{u,v}$ to represent the propagation probability along the edge. Note edge (u, v) is directional, v is an out-neighbor of u and u is an in-neighbor of v. There are different diffusion models which can be used to define influence propagation process. Without loss of generality, we adopt the Independent Cascade (IC) diffusion model [1,2]. Initially, every node is inactive. If a node u is selected as a seed, u becomes active and attempts to activate one of its inactive out-neighbors. The newly activated nodes will attempt to activate their inactive out-neighbors. Regardless of success or not, the same node will never get second chance to activate the same inactive out-neighbor. This process terminates when no more inactive nodes can be activated. In particular, we say a node v is successfully activated by a set S of seeds if and only if the overall influence from S to v is above a given threshold. In addition, the success of node u in activating out-neighbor v is determined via the maximum influence path [43].

In social networks, the subset of nodes $S \subseteq V$, which are active initially before influence propagation process starts, are known as *seeds*. Each seed spreads information to inactive nodes in the social networks. For an inactive node v, we define the aggregated probability that v is activated by the seeds in S:

Definition 1 (*Aggregated Influence Probability*)**.**

$$Pr(v|S) = 1 - \prod_{u \in S}(1 - Pr(p^{max}_{u,v})). \tag{1}$$

where $p^{max}_{u,v}$ is the maximum influential path from u to v. Suppose $p^{max}_{u,v}$ is $\{u, v_i, \dots, v_j, v\}$. $Pr(p^{max}_{u,v})$ is the probability that u can influence v along the path $p^{max}_{u,v}$, i.e., $Pr(p^{max}_{u,v}) = w_{u,v_i} \times \cdots \times w_{v_j,v}$. Since $p^{max}_{u,v}$ is the maximum influential path, $Pr(p^{max}_{u,v})$ is greater than that along any other path from u to v in the social network.

Definition 2 (*Activated Nodes*)**.** Given a set of seeds S, the node set $\sigma(S)$ is a subset of nodes in V and those nodes can be activated by S:

$$\sigma(S) = \bigcup_{v \in V, Pr(v|S) \geq \delta} \{v\}. \tag{2}$$

where δ is the activation threshold.

Definition 3 (*Influence Maximization (IM)*)**.** Given a social network $G = (V, E)$ and an integer k, the influence maximization is to find a set of nodes $S \subseteq V$, known as *seeds*, such that, if only the nodes in S are active initially, the number of nodes activated by S, at the end of information propagation process following IC diffusion model, is maximized, i.e.,

$$\arg\max_{S \subseteq V, |S| \leq k}\{|\sigma(S)|\}. \tag{3}$$

3.2. Community-diversified influence maximization

Given a set of seeds S, we define the community diversity of the nodes activated by S. As we know, the nodes in social networks can be grouped into different communities. If the activated users are from more communities, it implies the higher diversity. The community diversity is evaluated as follows:

Definition 4 (*Community Diversity Function*)**.** Suppose the nodes in a social network $G = (V, E)$ have been organized into m communities, denoted as $\mathbb{C} = \{C_1, \dots, C_m\}$. Given a set of seeds S, the diversity of nodes activated by S is defined as:

$$D(S) = \sum_{C_i \in \mathbb{C}} \sqrt{\sum_{v_j \in C_i \cap \sigma(S)} r(v_j)} \tag{4}$$

where v_j is a node activated by S (i.e., $v_j \in \sigma(S)$) and a member in community C_i, $r(v_j)$ represents the importance of v_j in social networks.

$D(S)$ is greater when the community diversity of activated nodes increases. Specifically, when activating a node from a new community (i.e., this community does not have any activated node yet), the higher score is awarded. For the nodes from the same community, the award for activating them decreases by applying the square root operator. The similar idea has been used in document summarization [44]. For node v_j, the importance in social networks $r(v_j)$ can be the degree of v_j, the PageRank score or the betweenness values of v_j, or any other user-defined score function. The different values of nodes' importance may help to discover the effective communities via considering the real influence of nodes and communities. But this is out of this research work. Therefore, in this work, $r(v_j)$ is set as 1 by default for the generalization.

Definition 5 (*Community-diversified Influence Maximization (CDIM)*)**.** Given a social network $G = (V, E)$ and an integer k, CDIM problem aims to find a set of seeds $S \subseteq V$ satisfying:

$$\phi(S) = \arg\max_{S \subseteq V, |S| \leq k}\{(1 - \lambda)\frac{|\sigma(S)|}{|V|} + \lambda\frac{D(\sigma(S))}{D(V)}\}. \tag{5}$$

where $\sigma(S)$ represents the set of nodes activated by S, $D(\sigma(S))$ represents the community diversity of $\sigma(S)$; $\lambda \in [0, 1]$ is the trade-off parameter to balance the two objectives, i.e., the number and the community diversity of the activated nodes; $|V|$ and $D(V)$ are the constants for normalization.

3.3. Monotone and submodularity

The evaluation metric $\phi(.)$ in Eq. (5) is monotonous and submodular. To prove this, we show that $|\sigma(.)|$ and $D(.)$ are monotonous and submodular respectively. Given any trade-off parameter $\lambda \in [0, 1]$, the aggregation function of two monoto nous and submodular functions is still monotonous and submodular.

Lemma 1. *$|\sigma(.)|$ is monotonous and submodular.*

The influence maximization using IC models has been proved (Theorem 2.2 in [2], Theorem 2 in [8]). it is not obvious for the adapted function $|\sigma(.)|$ to be true. Therefore, we summarize the proof as below.

Each social network can be treated as a random graph. Each edge $(u, v) \in E$ is associated with a random Bernoulli variable governed by $w_{u,v}$, which controls the likelihood u activates v. Let X denote the entire probability space constituting all possible determined influence propagation graphs. A determined influence propagation graph is generated by flipping a coin of bias $w_{u,v}$ for every edge $(u, v) \in E$ to determine if (u, v) exists in the determined graph. Then we have $Pr(v|S) = \sum_{x\in X} P(x)I(S, v, x)$, where $P(x)$ is the probability of a possible determined graph x, and $I(S, v, x)$ is an indicator to say if v can be reached from one of nodes in S in the determined graph x. If the indicator is true, then $I(S, v, x)$ equals 1. Otherwise, $I(S, v, x)$ equals 0. As $\sigma(S)$ is a node set $\bigcup_{v\in V, Pr(v|S)\geq\delta}\{v\}$ based on Eq. (2), the size of the node set $|\sigma(S)|$ is equivalent to $\sum_{v\in V}\{1|\sum_{x\in X} P(x)I(S, v, x) \geq \delta\}$.

We can safely say the function $|\sigma(.)|$ is monotone if the inequality $|\sigma(S \cup \{u\})| \geq |\sigma(S)|$ holds. To verify the inequality, let us comparing their alternatives $\sum_{v\in V}\{1|\sum_{x\in X} P(x)I(S \cup \{u\}, v, x) \geq \delta\}$ and $\sum_{v\in V}\{1|\sum_{x\in X} P(x)I(S, v, x) \geq \delta\}$. Here, if $I(S, v, x)$ equals 1, i.e., v is reachable from S in the determined graph x, then $I(S \cup \{u\}, v, x)$ must be 1. Conversely, it does not hold, i.e., if $I(S \cup \{u\}, v, x)$ is 1, then $I(S, v, x)$ may be 0 or 1. Since $P(x) \in (0, 1]$, $\sum_{x\in X} P(x)\ I(., v, x)$ is monotonous. Thus, $\sum_{v\in V}\{1|\sum_{x\in X} P(x)\ I(S \cup \{u\}, v, x) \geq \delta\}$ is always no less than $\sum_{v\in V}\{1|\sum_{x\in X} P(x)\ I(S, v, x) \geq \delta\}$.

Let $S \subseteq T \subseteq V$, $u \in V$ and $u \notin T$. We first consider a determined graph $x \in X$. $\mathbb{R}_x(S \cup \{u\}) - \mathbb{R}_x(S)$ is the set of nodes reachable from u, but not reachable from S, in the determined graph x. As $S \subseteq T$, we have $\mathbb{R}_x(S \cup \{u\}) - \mathbb{R}_x(S)$ must have equal or more additional reachable nodes than $\mathbb{R}_x(T \cup \{u\}) - \mathbb{R}_x(T)$. Thus $|\mathbb{R}_x(.)|$ is a submodular function. Noticing that $|\sigma(.)|$ is a non-negative linear combination of submodular functions $\mathbb{R}_x(.)$ over the determined graph space X with the threshold δ. Thus, $|\sigma(.)|$ is also submodular. The proof of Lemma 1 is proved.

Lemma 2. *$D(.)$ is monotonous and submodular.*

Since $\sigma(.)$ is a monotone and submodular function to be proved in Lemma 1, we have $\Delta(u|S) \geq \Delta(u|T)$ for any $S \subseteq T \subseteq V$ and $u \in V \setminus T$ where $\Delta(u|S) = \sigma(S \cup \{u\}) - \sigma(S)$ representing the set of nodes that are activated by u, but not by S.

Since $D(S)$ is defined as $\sum_{i=1}^{m}\sqrt{\sum_{v_j\in C_i\cap\sigma(S)} r(v_j)}$, if we suppose $D_i = \sqrt{\sum_{v_j\in C_i\cap\sigma(S)} r(v_j)}$, then $D(S)$ can be expressed as $\sum_{i=1}^{m} D_i$. If we can prove that $D_i(.)$ is a monotone and submodular function, then $D(.)$ must possess the general submodular property. We can prove $D_i(.)$ being a monotone and submodular function by proving $\sum_{v_j\in C_i\cap\sigma(S)} r(v_j)$ monotonous and submodular because applying the square root to a monotone submodular function yields a submodular function, and summing them all together retains submodularity.

From $\Delta(u|S) = \sigma(S \cup \{u\}) - \sigma(S)$, we can get $\sum_{v_j\in C_i\cap\sigma(S\cup\{u\})} r(v_j) - \sum_{v_j\in C_i\cap\sigma(S)} r(v_j) = \sum_{v_j\in C_i\cap\Delta(u|S)} r(v_j)$. Similarly, we can get that $\sum_{v_j\in C_i\cap\sigma(T\cup\{u\})} r(v_j) - \sum_{v_j\in C_i\cap\sigma(T)} r(v_j) = \sum_{v_j\in C_i\cap\Delta(u|T)} r(v_j)$. Because $\Delta(u|S) \geq \Delta(u|T)$ holds, we have that $\sum_{v_j\in C_i\cap\Delta(u|S)} r(v_j) \geq \sum_{v_j\in C_i\cap\Delta(u|T)} r(v_j)$ for the same community C_i. Thus, for any $S \subseteq T \subseteq V$ and $u \in V \setminus T$, it can conclude that $\sum_{v_j\in C_i\cap\sigma(S\cup\{u\})} r(v_j) - \sum_{v_j\in C_i\cap\sigma(S)} r(v_j) \geq \sum_{v_j\in C_i\cap\sigma(T\cup\{u\})} r(v_j) - \sum_{v_j\in C_i\cap\sigma(T)} r(v_j)$. Therefore, we can see that $\sum_{v_j\in C_i\cap\sigma(S)} r(v_j)$ satisfies the submodular property. Obviously, it also satisfies the monotone property. Lemma 2 is proved.

4. Solution frameworks

This section proposes two solutions of CDIM problem.

4.1. Standard greedy approach

The monotone and submodularity property of $\phi(.)$ shown in Section 3.3 guarantees that the greedy algorithm of CDIM problem is with $(1 - \frac{1}{e} - \epsilon)$-approximation.

Algorithm 1 Greedy Algorithm

Input: A social network $G = (V, E)$, an integer k, an activation threshold δ, communities $\{C_1, ... C_m\}$
Output: A set of k nodes

```
Initialize $i = 1$, $S_0$ = NIL;
while $i \leq k$ do
    Initialize temporary variants $u_{best} = \emptyset$, Score$_{best} = 0$;
    for each node $u \in V \setminus S_{i-1}$ do
        $\Delta(u) = \phi(S_{i-1} \cup \{u\}) - \phi(S_{i-1})$;
        if $\Delta(u) \geq$ Score$_{best}$ then
            $u_{best} = u$;
            Score$_{best}$ = $\Delta(u)$;
    $S_i = S_{i-1} \cup \{u_{best}\}$;
    $i++$;
return $S_i$;
```

Algorithm 1 briefly states the procedure of the standard greedy algorithm. Suppose there are m communities in social networks. Initially, the seed set S is empty. The greedy algorithm runs by k iterations. At iteration i, if a node u leads to the maximal *community-diversified influence gain*, denoted as $\Delta(u)$, u is selected as a seed and inserted into S (denoted as S_{i-1} before inserting the new seed at iteration i). The community-diversified influence gain is defined as

$$\Delta(u|S_i) = \phi(S_{i-1} \cup \{u\}) - \phi(S_{i-1}). \tag{6}$$

In this work, $\phi(.)$ is calculated based on the sampling technique discussed in [1,2]. The time complexity of the greedy algorithm is $O(kn^2 \cdot \frac{1}{2\epsilon^2} log \frac{n}{\eta})$ where n is the number of nodes in the social network, ϵ and η are two sampling parameters in [1,2]. The complexity consists of two parts: the first one $O(kn)$ means that the algorithm needs to run k iterations and, at each iteration, it requires to probe each node in the social network; the second part $O(n \cdot \frac{1}{2\epsilon^2} log \frac{n}{\eta})$ means that the estimation of $\phi(S)$ needs to check each node in the social network to determine whether it can be activated in the sampled graphs of size $\frac{1}{2\epsilon^2} log \frac{n}{\eta}$. The error bound of the greedy algorithm is $(1 - \frac{1}{e} - \epsilon)$ where $(1 - \frac{1}{e})$ comes from the greedy approximation and ϵ comes from the sampling approximation.

4.2. Upper bound algorithm

To improve the efficiency of the greedy algorithm, we develop an upper bound based approach in order to reduce the unnecessary computations as much as possible. Next, we show the existence of upper bound.

Lemma 3. *Given any node u, if it is selected as a seed at one of k iterations, the community-diversified influence gain cannot exceed the community-diversified influence gain if u is the first selected seed.*

Proof. Since $\phi(.)$ has been proved to be monotonous and submodular in Section 3.3, we can derive that $\Delta(u|S_{i-1}) \geq \Delta(u|S_i)$ for any node $u \in V \setminus S_i$ where $S_{i-1} \subseteq S_i$. Let $\Delta_i(u)$ denote the community-diversified influence gain of u at iteration i. If $\Delta_i(u)$ is greater than $\Delta_i(v)$ for any $v \in V \setminus S_i$, $v \neq u$, u is selected as a seed at iteration i. Thus, we have $\Delta_{i-1}(u) \geq \Delta_i(u)$. It means that the community-diversified influence gain by selecting a node as a new seed in the earlier iterations must be not less than that by selecting it in the later iterations. In addition, it is easy to see $\Delta_0(u) = \phi(\{u\})$. So $\Delta_0(u)$ is the upper bound of the community-diversified influence gain by selecting u as a seed at any of the k iterations. □

According to Lemma 3, if the community-diversified influence gain of node u at iteration i, denoted as $\Delta_i(u)$, is known, we can safely prune any node if the upper bound of its community-diversified influence gain is less than $\Delta_i(u)$. Furthermore, the upper bounds provide the probing priority for the nodes not pruned. That is, the nodes with higher upper bounds should be evaluated earlier.

Besides using the sole influence of a node as an upper bound, we also explore the upper bound based on the intermediate computational results, which allow us to avoid computing comm unity-diversified influence gain of many nodes at each iteration and thus further improve the efficiency of the whole algorithm.

Lemma 4. *Suppose we have a seed candidate node $u*$ with community-diversified influence gain $\Delta_i(u*)$ at the ith iteration, and the seed set S_{i-1} has been identified. For any seed candidate node u, computing community-diversified influence gain of u can be avoided at the ith iteration if u satisfies*

$$\phi(S_{i-2} \cup \{u\}) + \phi(\{x_{i-1}\}) - \phi(S_{i-1}) \leq \Delta_i(u*). \tag{7}$$

where x_{i-1} is the selected seed node at the $(i-1)$th iteration.

Proof. We know

$$\begin{aligned}\Delta_i(u) &= \phi(S_{i-1} \cup \{u\}) - \phi(S_{i-1}) \\ &= \phi(S_{i-2} \cup \{x_{i-1}\} \cup \{u\}) - \phi(S_{i-1}) \\ &= \phi(\{S_{i-2} \cup \{u\} \cup \{x_{i-1}\}) - \phi(S_{i-1}) \\ &\leq \phi(\{S_{i-2} \cup \{u\}\}) + \phi(\{x_{i-1}\}) - \phi(S_{i-1}).\end{aligned} \tag{8}$$

Thus, for any node u, the upper bound of its community-diversi fied influence gain in the ith iteration can be estimated using $\phi(\{S_{i-2} \cup \{u\}\}) + \phi(\{x_{i-1}\}) - \phi(S_{i-1})$. If the upper bound is lower than or equal to the community-diversified influence gain $\Delta_i(u*)$ of an observed candidate $u*$ in the ith iteration, then u can be safely skipped without computing exact community-diversified influence gain. □

Algorithm 2 demonstrates the procedure of upper bound algorithm. At the beginning, we initialize the algorithm. In Line 5–Line 17, we run k iterations and select the best seed node at each iteration. In Line 7–Line 15, we only check the nodes having its sole influence $\phi(\{u\})$ larger than the maximal community-diversified influence gain of the currently observed nodes (Lemma 3). And then we check if it satisfies the condition specified in Lemma 4 at Line 8. If the node u can successfully pass the two filter conditions, then computing the community-diversified influence gain of u in Line 11–Line 15. Among all observed seed candidates, the maximal community-diversified influence gain is maintained by *maxMarGain*.

Algorithm 2 Upper Bound Algorithm.

Input: A social network $G = (V, E)$, an integer k, an activation threshold δ, communities $\{C_1, \dots C_m\}$.
Output: A set of k nodes

```
1:  Initialize $S_0$ =NIL, $\phi(S_0) = 0$, $x_0$ =NIL;
2:  for each node $u \in V$ do
3:    compute $\phi(\{u\})$;
4:    Record $(u, \phi(\{u\}))$ into a queue $Q_0$;
5:  for $i = 1 : k$ do
6:    maxMarGain = $-\infty$;
7:    for each node $u \in V \setminus S_{i-1}$ and $\phi(\{u\}) \geq$ maxMarGain do
8:      if $u \in Q_{i-1}$ and $(\phi(S_{i-2} \cup \{u\}) + \phi(x_{i-1}) - \phi(S_{i-1}))<$ maxMarGain then
9:        Do nothing;
10:     else
11:       Compute $\phi(S_{i-1} \cup \{u\})$;
12:       Record $(u, \phi(S_{i-1} \cup \{u\}))$ into a queue $Q_i$;
13:       if $\phi(S_{i-1} \cup \{u\}) - \phi(S_{i-1}) >$maxMarGain then
14:         maxMarGain = $\phi(S_{i-1} \cup \{u\}) - \phi(S_{i-1})$;
15:         $x_i = u$;
16:   $S_i = S_{i-1} \cup \{x_i\}$;
17:   $\phi(S_i) = \phi(S_{i-1})+$ maxMarGain;
18: return $S_i$;
```

5. Community-aware influence estimation

Given a set of seeds S and any node v_j in a community C_i, this section proposes an innovative method to efficiently determine whether v_j can be activated by S or not.

Consider a social network $G = (V, E)$ with propagation probability $w_{u,v}$ on edge $(u, v) \in E$. Let $p_{u,c}$ be any path from u to v and the sequence of nodes along the path is $\langle n_1, n_2, .., n_x\rangle$ where $n_1 \equiv u$ and $n_x \equiv v$. As introduced in Section 3, the probability that v is influenced by u through this path equals to the product of propagation probabilities on edges along this path, denoted as $Pr(p_{u,v}) = w_{n_1,n_2} \times w_{n_2,n_3} \times, \dots, \times w_{n_{x-1},n_x}$. As each node has only one chance to activate its neighbors, the best chance that v is influenced by u is through the *most influential path* from u to v, known as $p^{max}_{u,v}$, i.e., the path with the maximum influence probability as introduced in Section 3.

Given a set of seeds S and a node v, the aggregated influence probability from S to v can be evaluated following Eq. (1), i.e., through the most influential paths starting from all seeds in S to v.

Path transformation

Given a path from u to v in a social network G, the influence probability is $Pr(p_{u,v}) = w_{n_1,n_2} \times w_{n_2,n_3} \times, \dots, w_{n_{x-1},n_x}$. Instead of computing $Pr(p_{u,v})$, we compute $\log(Pr(p_{u,v})) = \log(w_{n_1,n_2}) + \log(w_{n_2,n_3}), \dots, +\log(w_{n_{x-1},n_x})$. If w_{n_i,n_j} for every edge (n_i, n_j) in E is transformed to $-\log(w_{n_i,n_j})$,[1] the problem searching for $p^{max}_{u,v}$ is transformed to search for the shortest path from u to v. Once the shortest path $p^{max}_{u,v}$ is obtained, $Pr(p^{max}_{u,v}) = e^{(p^{max}_{u,v}.dist)}$ where $p^{max}_{u,v}.dist$ is the shortest path distance.

[1] Since w_{n_i,n_j} is in (0,1], $\log(w_{n_i,n_j})$ is negative. The most influential path is the path with the maximum value. By changing the sign from $\log(w_{n_i,n_j})$ to $-\log(w_{n_i,n_j})$, the most influential path is equivalent to the path with the minimum distance.

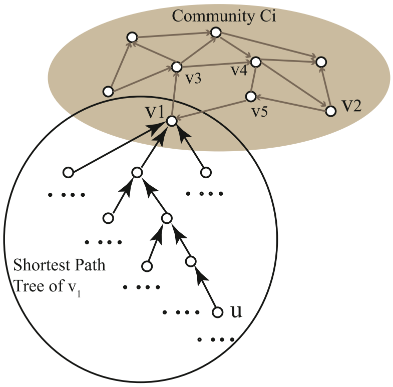


Fig. 1. Community-aware Partial Shortest Path Tree.

Community-aware Partial Shortest Path Tree (CPSP-Tree)

Suppose the path transformation has been done. To estimate the influence probability of a seed set S to any node v, a shortest path tree can be constructed for every node u in the social network G where the root is v. For any node u in the tree, the path to root corresponds to the shortest path from u to v, i.e., $p^{max}_{u,v}$. So, the influence probability from u to v (i.e., $Pr(p^{max}_{u,v})$) can be directly calculated. The shortest path tree is the compressed version of the shortest paths from all other nodes in G to v by merging the same node appearing in different paths. Given a set of seeds S and a node v, the aggregated influence probability of S to v can be computed by finding all seed nodes in S in the tree; for each seed node, the influence probability to v is calculated, then we compute the aggregated influence following Eq. (1).

Building and maintaining complete shortest path trees is time and space consuming. To handle this issue, we propose Commu nity-aware Partial Shortest Path Tree. For each community C_i, a node is selected as the community representative v_{C_i}. Note, for community C_i, we only build and maintain one shortest path tree, i.e., the shortest path tree of the representative node v_{C_i}. Given any node u in G, the shortest path distance from u to v_{C_i} can be retrieved in the shortest path tree; the shortest path distance from u to any other node in C_i cannot be obtained from the shortest path tree but the upper and lower bound can be estimated using triangle inequality. Fig. 1 shows an example. p^{max}_{u,v_1} can be obtained from the shortest path tree of v_1 directly; the distance of p^{max}_{u,v_2} is bounded by $p^{max}_{u,v_1}.dist + p^{max}_{v_1,v_2}.dist$ and $p^{max}_{u,v_1}.dist + p^{max}_{v_2,v_1}.dist$ where p_{v_1,v_2} goes through v_1, v_3, v_4, v_2 and p_{v_2,v_1} goes through v_2, v_5, v_1.

Based on the general community metrics, the intra-comm unity members often have close relationships/less number of hops than the inter-community members, e.g., for clique-like communities, the intra-community members has 1 hop. Of course, for the other types of communities, there may be different between the distance of intra-community members and that of inter-community members. Motivated by this, if v_1 and v_2 are in the same community, in general the shortest distance from v_1 to v_2 is typically much smaller that of u and v_2 where u and v_2 are from different communities. So the shortest distance could be the path from v_2 to v_1, not u. Thus, $p^{max}_{u,v_1}.dist + p^{max}_{v_1,v_2}.dist$ and $p^{max}_{u,v_1}.dist + p^{max}_{v_2,v_1}.dist$ are reasonably tight upper and lower bound of $p^{max}_{u,v_2}.dist$. That is, a reasonably tight upper bound and lower bound of $Pr(p^{max}_{u,v_2})$ are known, denoted as $Pr(p^{max}_{u,v_2}).UB$ and $Pr(p^{max}_{u,v_2}).LB$.

Given a set of seeds S and any node v_j in any community C_i, the upper bound and lower bound of $Pr(v_i|S)$ are

$$\begin{aligned} Pr(v_j|S).UB &= 1 - \prod_{u\in S}(1 - Pr(p^{max}_{u,v_j}.UB)).\\ Pr(v_j|S).LB &= 1 - \prod_{u\in S}(1 - Pr(p^{max}_{u,v_j}.LB)). \end{aligned} \tag{9}$$

where $Pr(p^{max}_{u,v_j}.UB)$ and $Pr(p^{max}_{u,v_j}.LB)$ are obtained by exploring the shortest path tree of v_{C_j} as discussed.

Compared with the activation threshold δ, if $Pr(v_j|S).UB < \delta$, v_j is pruned since it cannot be activated; if $Pr(v_j|S).LB > \delta$, v_j must be activated. Only in the situation $Pr(v_j|S).UB \geq \delta \geq Pr(v_j|S).LB$, the exact $Pr(v_j|S)$ is computed by searching the shortest paths from all seeds in S to v_j.

The data structure for maintaining shortest paths in large graphs is a well studied field (e.g., [45]) where one approach is the shortest path tree. While the variants of shortest path tree have been developed for different purposes (e.g., [46]), the proposed CPSP-tree links each shortest path tree with the community information of the root so as to reduce the number of shortest path trees maintained. Note the algorithm to build shortest path trees is orthogonal to this study.

For each community C_i, one shortest path tree is built. There are m communities. The storage of all shortest path trees is $m|V|$. In community C_i, the shortest distances from (to) each member node to (from) the representative node v_{C_i} are precomputed and maintained; the storage is $2|V_{C_i}|$ where V_{C_i} is set of nodes in community C_i. The storage of all communities is $\sum_{i=1}^{m}|V_{C_i}|$. The overall storage of CPSP-Tree is $m|V| + \sum_{i=1}^{m}|V_{C_i}|$ and the complexity is $m|V|$.

The time complexity of constructing the CPSP-tree is analyzed as follows. Constructing the CPSP-tree is same to compute the single-source shortest path for each node in the graph. As we know, the single-source shortest path computation takes $\mathcal{O}(m + n \cdot \log\log n)$, which has been reported in [47]. In the worst case, we need to invoke the single-source shortest path computation for every node in the graph, i.e., $O(n)$ times. Then, the time complexity becomes $\mathcal{O}(n\cdot(m+n\cdot\log\log n))$. However, in practice, the CPSP-tree construction only work on the boundary nodes in the communities. The number n' of boundary nodes is much less than the total number n of nodes in the graph. As such, the time complexity is $\mathcal{O}(n' \cdot m + n' \cdot n \cdot \log\log n)$.

6. Unknown community based diversified influence spread

In the above sections, we developed novel index and solutions for addressing the community-aware influence maximization with diversification, i.e., we assume that the communities in the social network are known in advance. However, sometimes the community information may not be available. In this case, the most challenging part in this proposed problem is that we do not have clear guidance for selecting seeds, i.e., given a seed candidate, how diverse its influence spread to make it being selected as a real seed? To address this challenge, in this section we propose a heuristic approach to resolve the problem of diversifying influence spread without community information.

6.1. Diversity diminished model

Before the first seed to be selected, any node in the social network is treated equally in diversity. So we can select the first seed with the maximum influence spread without considering the factor of diversity. Assume the first seed u has been determined. Let $S = \{u\}$ be the current seed set and $\sigma(u)$ be the influenced nodes of u in the social network.

Definition 6 (*Possible Hits of a Node*)**.** Given a seed node u and its influenced node set $\sigma(u)$, the possible hits of u consists of the nodes $v \in V \setminus \sigma(u)$ satisfying that v has an edge to a node in $\sigma(u)$. The possible hits of u is denoted as $PH(u)$.

Similarly, the definition of the possible hits for a node can be applied to the seed set S, which is denoted as $PH(S)$ in this work. The intuition of our defined possible hits for seed nodes mainly comes from the fact that such nodes of possible hits have high probability of being activated when we add a new seed and the new seed is selected in the nearby regions of the existing seeds in S. This fact also matches with the IC diffusion model because a node can be jointly activated by different paths from multiple other nodes.

Now, let us introduce the objective function to push the selection of next seed far from the current possible hits as much as possible. By doing this, we guarantee that the selection of next seed is favorable to the node that has the maximum marginal gain regarding the currently selected seeds and the maximum marginal gain would be further discounted or diminished by the possible hits it can reach.

Definition 7 (*Possible Hits Biased Diminish*)**.** Given the possible hits $PH(S)$ of a seed set S, a seed candidate u' and its influenced node set $\sigma(u')$, the possible hits biased diminish giving the marginal gain $\Delta(u')$ is estimated by

$$Dim(u'|S) = \begin{cases} \frac{1}{log(|PH(S) \cap \sigma(u')|)}, \text{ if } |PH(S) \cap \sigma(u')| > 1 \\ 1, \text{ otherwise} \end{cases} \tag{10}$$

Definition 8 (*Diminished Objective Function*)**.** The diminished marginal gain of a seed candidate can be estimated by $Dim(u'|S_{i-1}) * \Delta(u')$ where $\Delta(u') = \sigma(S_{i-1} \cup \{u'\}) - \sigma(S_{i-1})$.

When the community information is not available, we approximate the diversified influence spread of the candidate nodes by computing the maximum benefit of those nodes via the diminished objective function. Obviously, the diminished objective function is not monotonic because $PH(S)$ may become larger or smaller with the increase of S, which leads that $Dim(.)$ is not a monotonic function, too. Therefore, the above frameworks are not directly applicable to the problem of diversified influence maximization without community information.

6.2. Local-optimality based approach

In this section, we propose the local optimality based approach to address the diversified influence spread problem when the community information is unknown in advance. The local optimality based approach is developed by adjusting the upper-bound based approach in Section 4.2.

Now let us analyze the upper bound and lower bound of nodes in the context of the diminished objective function, i.e., when a node can be determined to be a seed node based on the local information.

From Definition 8 and Eq. (10), we can easily get that given a new seed candidate u' and the current seed set S, its expected maximum marginal gain is in the range of $[\frac{\Delta(u')}{log(|PH(S) \cap \sigma(u')|)}, \Delta(u')]$ where $\frac{\Delta(u')}{log(|PH(S) \cap \sigma(u')|)}$ is the lower bound of the maximum marginal gain of u' and $\Delta(u')$ is its upper bound. As $log(|PH(S) \cap \sigma(u')|) \leq \min\{log|PH(S)|, log|\sigma(u')|\}$ always holds, we can relax the lower bound to $\frac{\Delta(u')}{\min\{log|PH(S)|, log|\sigma(u')|\}}$, which can be rewritten as $\max\{\frac{\Delta(u')}{log|PH(S)|}, \frac{\Delta(u')}{log|\sigma(u')|}\}$. Note that here $\Delta(u')$, $PH(S)$, and $\sigma(u')$ are the intermediate results in the process of running upper bound based approach. So we can quickly obtain the lower bound and upper bound of nodes and filter the insignificant nodes, by which we can reduce the computational cost.

Property 1 (*Local Lower Bound*)**.** *The local lower bound of u' at the ith iteration is measured by $\max\{\frac{\Delta(u')}{log|PH(S_{i-1})|}, \frac{\Delta(u')}{log|\sigma(u')|}\}$. Since it always has $\sigma(u') \geq \Delta(u')$, the local lower bound of u' can be further improved as $\max\{\frac{\Delta(u')}{log|PH(S_{i-1})|}, \frac{\Delta(u')}{log|\Delta(u')|}\}$.*

Property 2 (*Local Upper Bound*)**.** *The local upper bound of u' at the ith iteration is $\sigma(u')$, i.e., assuming there is no overlap between $PH(S_{i-1})$ and $\sigma(u')$. If $\Delta(u')$ has been computed, then $\Delta(u')$ will replace $\sigma(u')$ to be the tight local upper bound of u' because $\Delta(u') \leq \sigma(u')$ is true in all iterations.*

The key idea of the local-optimality based approach is to find the terminating node using the above local lower bound and upper bound properties. In other words, for those nodes afterwards the terminating node, they must not generate the seed node for the current iteration. By doing this, we can safely and quickly locate the best seed node at an iteration without probing all possible nodes.

Algorithm 3 Local-Optimality based Algorithm

```
Input: A graph $G = (V, E)$ and an integer $k$
Output: $S$ - the $k$-vertex set
1:  {Initialize the upper bound value for each node};
2:  for each node $u \in V$ do
3:      Write $u : \sigma(u)$ into an ordered queue $Q$;
4:  $S_1$ ←Q.pop(), i.e., $u_{best} : \sigma(u_{best})$;
5:  Set $i = 2$;
6:  while $i \leq k$ do
7:      Initialize an iteration: found = false, a seed candidate $s$ and $s$.value
        = 0;
8:      change $Q'$ to an element s;
9:      $u_{first} : \Delta(u_{first}) \leftarrow Q$.pop();
10:     $\Delta(u_{first})$ = $\sigma(S_{i-1} \cup \{u_{first}\})$ - $\sigma(S_{i-1})$;
11:     while !found && $Q$.getFirstUnvisited()≠ null do
12:         $u_{next}$:$\Delta(u_{next}) \leftarrow Q$.getFirstUnvisited();
13:         if $\max\{\frac{\Delta(u_{first})}{log|PH(S_{i-1})|}, \frac{\Delta(u_{first})}{log|\Delta(u_{first})|}\} \geq \Delta(u_{next})$ or $\frac{\Delta(u_{first})}{log|PH(S_{i-1}) \cap \Delta(u_{first})|} \geq$
            $\Delta(u_{next})$ then
14:             $S_i$ = $S_{i-1} \cup Q'$.pop();
15:             found = true;
16:         else
17:             $Q'$.add($u_{first} : \frac{\Delta(u_{first})}{log|PH(S_{i-1}) \cap \Delta(u_{first})|}$);
18:             $Q$.add($u_{first} : \Delta(u_{first})$);
19:     if !found then
20:         $S_i$ = $S_{i-1} \cup Q'$.pop();
21: return $S_i$;
```

Algorithm 3 presents the detailed procedure of this approach. Here, we calculate the general influence spread of nodes as their loose upper bounds, as shown in Line 2–3. Those nodes and their influence spreads are maintained in an ordered queue Q. Obviously, the first seed is the node with the maximum value in Q. The challenging part is how to select the other $k - 1$ best seed nodes satisfying our diminished objective function w.r.t. our problem of the diversified influence maximization. To address this, we first pop a node u_{first} and its value $\Delta(u_{first})$ from Q and compute its incremental gain regarding the selection S_{i-1}. Note now we did not consider the factor of diversification. And then, the second While-Loop is used to identify the terminating node in Q. We use the new queue Q' to maintain the nodes to be visited and the nodes' diminished value. In the second While-loop shown in Line 11–18, we check the unvisited nodes (e.g., say u_{next}) in Q and compare its upper bound value (e.g., $\Delta(u_{next})$) with the lower bound value (e.g., $\max\{\frac{\Delta(u_{first})}{log|PH(S_{i-1})|}, \frac{\Delta(u_{first})}{log|\Delta(u_{first})|}\}$) or its exact value (e.g., $\frac{\Delta(u_{first})}{log|PH(S_{i-1}) \cap \Delta(u_{first})|}$) of u_{first}. If the lower bound or the exact value of u_{first} is not less than the upper bound of an unvisited node u_{next}, then we can safely say u_{next} is the terminating node and take the node with the maximum value as the new seed from Q'. Otherwise, we add the intermediate results of u_{first} and its diminished value $\frac{\Delta(u_{first})}{log|PH(S_{i-1})\cap\Delta(u_{first})|}$ into the temporary queue Q', while write u_{first} and its updated value $\Delta(u_{first})$ back to Q. In Line 13, if the first condition is true, then we do not calculate the exact value, i.e., avoiding the operation of set merge. In the worst case, if no terminating node exists, i.e., all the nodes in Q have been probed, then the best seed should be the node with the maximum value in Q' because all their diminished values are the exact value in Q', shown in Line 19–20. Finally, the k best seeds will be returned.

6.3. Optimizing precision of local-optimality

Since the above local-optimality based approach selects the seed nodes based on the diminished objective function, the final solution will depend on the first seed node selection. Therefore, it can only produce the local optimum solution to our problem. For instance, there is a large community and a set of medium-sized communities around the large community. In this case, if we select the first seed node activating a large number of users in the large community, then it will make us to ignore the consideration of the groups of medium-sized communities even if their combination can contribute more than the only selection of the large community.

In general, this challenging issue often occurs in the problems of local search optimization. To do this, one way is to do iterate local search multiple times. Each time starts from a different initial configuration. This is called as repeated local search. However, how to select the different initial configurations is a tough question. To address this challenge, in this work we develop a heuristic approach to optimize the benefit of the extra search by utilizing the knowledge obtained during the previous local search phases. The key idea is to identify the first two seed nodes as the main line of local search using Algorithm 3. And then, we select the nodes as the initial seed candidates for other lines of local search where each node should have high possibility to drive a local search by checking if its independent influence spread value is larger than a certain ratio (i.e., $\lambda = 0.3$) of that of the first two seeds. As such, besides the main local search, we can also initialize different local search with the potential candidates at the beginning. In order to balance the optimized global precision and the efficiency of the heuristic approach, we have an assumption that we do not allow overlapped seed candidates appearing across any two lines of local search. When the algorithm continues to run, lots of lines of local search would be removed, by which the search space can be reduced soon.

The core procedure of the heuristic approach is similar to Algorithm 3. Therefore, we do not provide the pseudo codes in this paper.

7. Experimental study

We have conducted extensive experimental study to evaluate the effectiveness and efficiency of the proposed solution of CDIM problem. All these experiments are tested on a Red Hat Enterprise Linux Server (7.2), with 792GB RAM and Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GH shared by staff in the School of Science, RMIT University. The algorithms are implemented using Python 2.7.

Five real-world social network datasets are used in test. They are downloaded from *Stanford Large Network Dataset Collection*.[2] The statistics of the datasets are shown in Table 1.

Table 1
Statistics of datasets.

Data sets	#nodes	#Edges	Avg degree
Facebook	4,039	88,234	21.8
Citation (DBLP)	4,558	217,984	47.8
Gowalla	69,097	351,452	5.1
Youtube	52,675	636,864	12.1
Amazon	317,194	1,745,870	5.5

7.1. Evaluation of effectiveness

The objective of CDIM problem is to find k nodes in a social network which are capable to activate the maximum number of nodes communities from as diverse as possible. The effectiveness tests check whether the proposed solutions show advantage towards the objective. For this purpose, we have implemented the following four algorithms.

- *IM* is the solution of influence maximization problem [1,2]. As a baseline, it is used to solve CDIM problem. Given a seed u, the set of activated nodes by u is denoted as *Inf(u)*. For another node u', if it is selected as a seed, the new nodes activated by u' is *Inf(u')/Inf(u)*. For the node selected as the next seed, it must be able to activate the maximum number of new nodes in social networks.
- *DIV(LPA)* represents our proposed CDIM solution where the widely-accepted community detection method *LPA* [29] is applied to identify all communities in social networks.
- *DIV(Louvain)* represents our CDIM solution where another widely-accepted community detection method *Louvain* [48] is applied to identify all communities in social networks.
- *Heuristic* represents our CDIM solution where the community definition is not specified and thus the influence-based communities in social networks are considered as discussed in Section 6.

The performances are reported when k changes. By default, λ in Eq. (5) is 0.5 and activation threshold δ in Eq. (2) is 0.2.

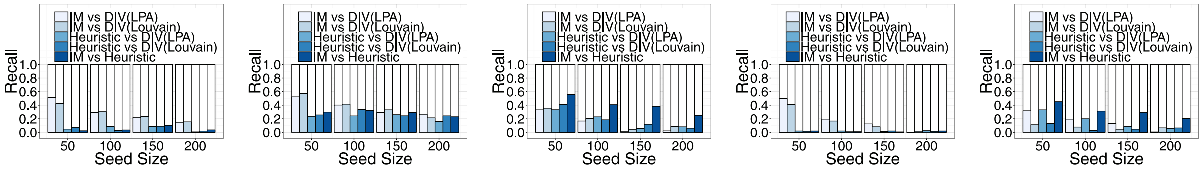
Precision

The precision is measured by comparing the seeds and the activated nodes returned by *IM* and *Heuristic* against those returned by *DIV(LPA)* and *DIV(Louvain)* respectively. Let the seed set (or the set of activated nodes) returned by *IM* be *im* and the seed set (or the set of activated nodes) returned by *DIV(.)* be div. The precision of *IM* vs. *DIV(LPA)* is $\frac{|im \cap div|}{|im|}$. If the precision is smaller, only a smaller fraction of IM problem solution overlaps CDIM solution; it implies IM problem is less similar to CDIM problem. Let the seed set (or the set of activated nodes) returned by *Heuristic* be *heu*. The precision of *Heuristic* vs. *DIV(LPA)* is $\frac{|heu \cap div|}{|heu|}$. The higher precision means influence-based communities can better approximate the specified communities.

For each data set, the seed sets returned by *IM*, *Heuristic*, *DIV(LPA)* and *DIV(Louvain)* are compared, and the test results are presented in Fig. 3 where k varies from 50 to 200. In Facebook, *IM* can identify 30%–50% of seeds returned by *DIV(.)* when k is 50 or 100. When k is 150 or 200, *IM* can identify about 20% of seeds returned by *DIV(.)*. In all settings of k, *Heuristic* can identify about 10% of seeds returned by *DIV(.)*. The similar trend can be observed for other three datasets.

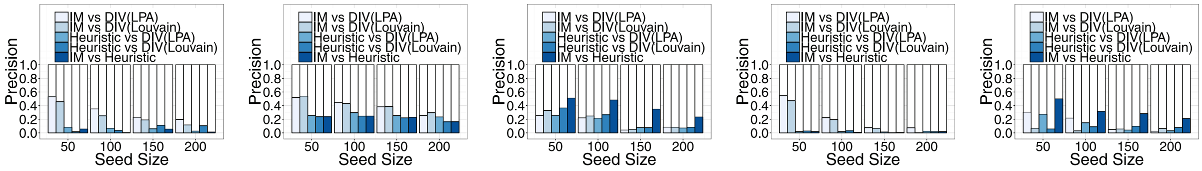
Similar to seed sets, the activated node sets using *IM* and *Heuristic* are compared with those using *DIV(LPA)* and *DIV(Louvain)* respectively. The test results are presented in Fig. 4. When k is 50 or 100, the precision of *IM* and *Heuristic* for Facebook, Citation and Youtube reaches 50%; the precision is about 20%–40% for Gowalla and about 30% for Amazon. When k

[2] http://snap.stanford.edu/data/index.html



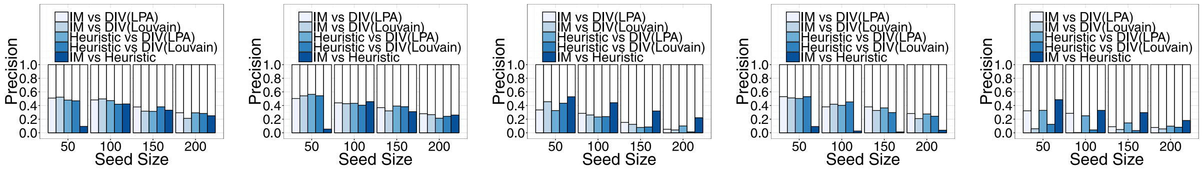
(a) Facebook-Seeds (b) Citation-Seeds (c) Gowalla-Seeds (d) Youtube-Seeds (e) Amazon-Seeds

Fig. 2. Recall of seeds.



(a) Facebook-Seeds (b) Citation-Seeds (c) Gowalla-Seeds (d) Youtube-Seeds (e) Amazon-Seeds

Fig. 3. Precision of seeds.



(a) Facebook-Influencees (b) Citation-Influencees (c) Gowalla-Influencees (d) Youtube-Influencees (e) Amazon-Influencees

Fig. 4. Precision of nodes activated.

is 150 or 200, the precisions are becoming much smaller for all data sets.

The test results clearly indicate that CDIM problem is significantly different from IM problem. Also, it indicates the influence-based communities are reasonable approximation of the communities applied in *DIV(LPA)* and *DIV(Louvain)*.

Recall

The recall is measured by comparing the seeds and the activated nodes returned by *IM* and *Heuristic* against those returned by *DIV(LPA)* and *DIV(Louvain)* respectively. The recall of *IM* vs. *DIV(.)* is defined as $\frac{|im \cap div|}{|div|}$. If the recall is smaller, the solution of IM problem is a smaller fraction of the solution of CDIM problem; it implies IM solution is less effective to CDIM problem. The recall of *Heuristic* vs. *DIV(.)* is defined as $\frac{|heu \cap div|}{|div|}$. The smaller recall means the influence-based communities are less effective to approximate the specified communities in DIV(.).

The recall in terms of seeds is presented in Fig. 2. For all five data sets, the recall of *IM* is lower than 55% and the recall of *Heuristic* is lower than 30% except Citation dataset. The recall in terms of activated nodes is presented in Fig. 5. In the dense datasets Facebook and Citation, *IM* can activate about 50% nodes among all nodes activated by *DIV(.)*. *Heuristic* can activate about 40% except *k*=50. For the other three datasets, *IM* and *Heuristic* have less recall values. The test results in recall lead to the similar conclusion as the test results in precision.

Community diversity

A community is influenced if at least one node of the community is activated. The real world community information of datasets Amazon and Youtube are known. Using *DIV(LPA)* to solve CDIM over the two datasets. The tests results are reported in Table 2 and Table 3 respectively. Obviously, the trade-off between the number of communities influenced and the number of nodes activated can be adjusted by setting λ different values (see Eq. (5)). When $\lambda = 0$, CDIM is degraded to influence maximization problem. As a result, the number of nodes activated is the highest while the number of communities influenced is the lowest. When λ increases, the number of nodes activated keeps decreasing while the number of communities influenced keeps increasing; the CDIM score (i.e., $\phi(S)$ for the returned seed set S) keeps increases.

Table 2
Activated nodes and influenced communities on Amazon dataset ($k = 200$, $\delta = 0.9$).

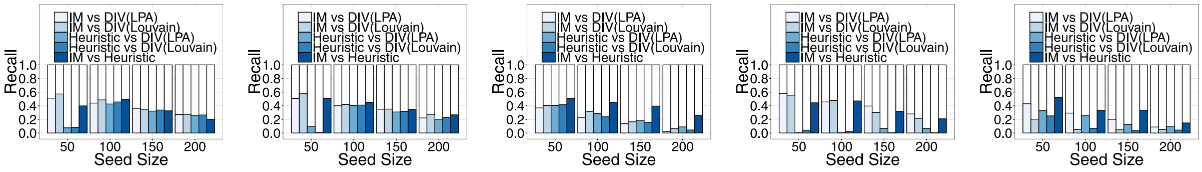
	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 0.75$
Communities influenced (#)	1704	1843	1938	**2031**
Communities influenced (%)	2.27	2.45	2.57	**2.69**
Influence (#nodes)	**2197**	2188	2132	1966
Influence (%)	**0.69**	**0.69**	0.67	0.62
Diversity score	3037.16	3190.20	3218.91	**3242.26**
Diversity score (normalized)	0.013	0.014	0.014	**0.014**

Table 3
Activated nodes and influenced communities on Youtube dataset ($k = 200$, $\delta = 0.5$).

	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 0.75$
Communities influenced (#)	1072	2382	2561	**2741**
Communities influenced (%)	6.54	14.54	15.63	**16.73**
Influence (#nodes)	**40,501**	40,447	40,427	40,365
Influence (%)	**76.89**	76.79	76.75	76.63
Diversity score	6246.62	9009.97	9180.49	**9340.65**
Diversity score (normalized)	0.18	0.25	**0.26**	**0.26**

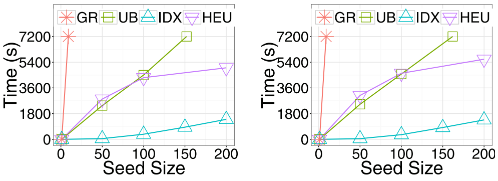
7.2. Evaluation of efficiency

The efficiency of *GR* (Algorithm 1), *UB* (Algorithm 2), *IDX* (i.e., *UB* with support of CPSP-Tree) at different settings of *k* are tested. Figs. 6–10 reports the time consumed using different



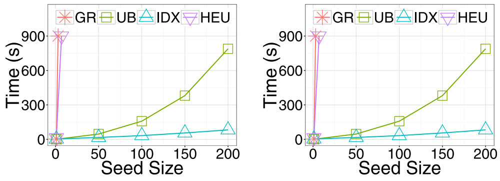
(a) Facebook-Influencees (b) Citation-Influencees (c) Gowalla-Influencees (d) Youtube-Influencees (e) Amazon-Influencees

Fig. 5. Recall of nodes activated.



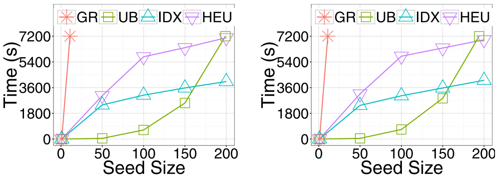
(a) Facebook(LPA) (b) Facebook(Louvain)

Fig. 6. Efficiency of proposed algorithms in Facebook dataset.



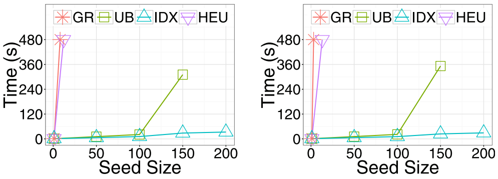
(a) Amazon(LPA) (b) Amazon(Louvain)

Fig. 10. Efficiency of proposed algorithms in Amazon dataset.



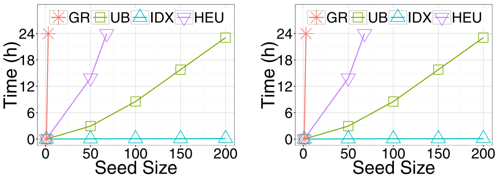
(a) Citation(LPA) (b) Citation(Louvain)

Fig. 7. Efficiency of proposed algorithms in DBLP dataset.



(a) Gowalla(LPA) (b) Gowalla(Louvain)

Fig. 8. Efficiency of proposed algorithms in Gowalla dataset.



(a) Youtube(LPA) (b) Youtube(Louvain)

Fig. 9. Efficiency of proposed algorithms in Youtube dataset.

algorithms for solving CDIM problem in different datasets. The communities are detected using LPA and Louvain. Note the reported time does not include the time consumed for community detection since it is orthogonal to CDIM problem. That is, community detection is performed offline. Once the communities are known, CPSP-Tree is built offline as well.

As shown in Figs. 6–10, *IDX* outperforms the other two algorithms by 8–40 times on all datasets (except Citation). In Fig. 6, *GR* performance is the worst in all situations. In Fig. 7, *UB* is better than *IDX* when k is 50, 100, and 150. But when k increases up to 160, *IDX* performs much better than *UB*. When k is 200, *IDX* is about 2.5 times faster than *UB*. This is because Citation is a dense social network. The dense social network tends to have more computation for intra-community nodes. Additionally, we observe that each algorithm performs similarly over the same data set with LPA and Louvain.

7.3. Other evaluations

This section presents the test results when changing the values of trade-off parameter λ, activation threshold δ, and size of dataset.

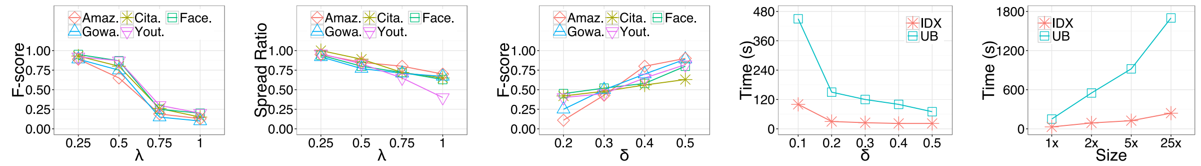
Varying λ

The accuracy of the activated nodes using *DIV(LPA)* and *IM* is evaluated respectively. The accuracy is measured by F-measure where $F = 2 * \frac{precision*recall}{precision+recall}$. The higher F-measure means the activated nodes are from more different communities. The test results are shown in Fig. 11(a) when λ varies from 0.25 to 1. In the situation $\lambda = 1$, it means the number of communities influenced is the sole factor considered. When $\lambda = 0.25$, it means the number of communities influenced is considered less while the number of nodes activated is considered more. By default, $k = 100$ and $\delta = 0.2$. F-score decreases with the increase of λ for all five data sets. And the decreasing trend goes slowly before $\lambda = 0.5$ and goes sharply after $\lambda = 0.5$.

IM aims to activate the maximum number of nodes in social networks and *DIV(LPA)* aims to activate maximum number of nodes which are from more communities. Compared to *IM*, *DIV(LPA)* has to sacrifice a certain number of activated nodes to ensure more communities are influenced. The ratios between the number of nodes activated using *DIV(LPA)* and that using *IM* at different settings of λ are reported in Fig. 11(b). When λ is between 0.25 and 0.75, the number of nodes activated using *DIV(LPA)* is 75% of that using *IM* for all datasets. When λ is 1, the ratio is 75% in four datasets and 40% in Youtube.

Varying δ

In the tests, $k = 100$ and $\lambda = 0.5$. As shown in Fig. 11(c), F-score increases when δ varies from 0.2 to 0.5. The results depict that it is easy for *IM* to activate more nodes when the higher value is set to δ. We also evaluate the time consumed by different algorithms over Amazon when δ changes. The test results are shown in Fig. 11(d). The performance of *UB* is accelerated for the higher δ value, but the performance of *IDX* is affected slightly by δ. Clearly, *IDX* always outperforms *UB*.



(a) F-Score of IM vs. DIV(LPA)

(b) Influence ratio of DIV(LPA) vs. IM

(c) F-Score of IM vs. DIV(LPA)

(d) Time cost with δ

(e) Varying data size

Fig. 11. Impact of λ, δ and data size.

Varying size of dataset

We generate three synthetic datasets based on Amazon in order to evaluate the scalability of the proposed algorithms for datasets of different sizes. The three datasets have 2 times, 5 times and 25 times of nodes in the original Amazon dataset shown in Table 1. Fig. 11(e) illustrates *UB* consumes 600 s when data size is 2 times and it consumes 1800 s when the data size is 25 times. Obviously, while the data size increases by about 12 times, the consumed time increases by 3 times only. *IDX* performs much better and the trending slope is very gentle.

8. Conclusion

This paper has studied community-diversified influence maximization (CDIM) problem. CDIM provides an innovative perspective to evaluate the influence propagation over social networks. The objective is to find *k* influential nodes in social networks as seeds such that, if a message is initiated by the seeds, the number of activated nodes as well as the number of communities to which the activated nodes belong can be maximized at the end of propagation process. The community diversity can reduce the risk of market campaigns and make more future impact. The goodnesses of CDIM solution are evaluated with a metric which have been proven monotonous and submodular. It allows greedy algorithm to be applied with reasonable approximation bound. To enable more efficient processing, the upper bound of community-diversified influence has been explored to minimize the seed candidates. In particular, an innovative CPSP-Tree index has been developed to quickly identify seed candidates. The intrinsic communities of CDIM are explored in the case that the communities cannot be provided by users. The extensive tests on five real-world datasets have verified the superiority of proposed solutions.

Acknowledgments

This work was mainly supported by the ARC Linkage Projects under Grant No. LP180100750.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25–28, 2010, 2010, pp. 1029–1038.

[2] D. Kempe, J.M. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24–27, 2003, 2003, pp. 137–146.

[3] N. Barbieri, F. Bonchi, G. Manco, Topic-aware social influence propagation models, in: ICDM, 2012, pp. 81–90.

[4] Ç. Aslay, N. Barbieri, F. Bonchi, R.A. Baeza-Yates, Online topic-aware influence maximization queries, in: Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24–28, 2014, 2014, pp. 295–306.

[5] S. Chen, J. Fan, G. Li, J. Feng, K. Tan, J. Tang, Online topic-aware influence maximization, Proc. VLDB Endow. 8 (6) (2015) 666–677.

[6] M. Gomez-Rodriguez, D. Balduzzi, B. Schölkopf, Uncovering the temporal dynamics of diffusion networks, in: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, 2011, pp. 561–568.

[7] M. Gomez-Rodriguez, B. Schölkopf, Influence maximization in continuous time diffusion networks, in: ICML, 2012.

[8] B. Liu, G. Cong, D. Xu, Y. Zeng, Time constrained influence maximization in social networks, in: 12th IEEE International Conference on Data Mining, ICDM 2012, 2012, pp. 439–448.

[9] G. Li, S. Chen, J. Feng, K. Tan, W. Li, Efficient location-aware influence maximization, in: International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22–27, 2014, 2014, pp. 87–98.

[10] X. Wang, Y. Zhang, W. Zhang, X. Lin, Distance-aware influence maximization in geo-social network, in: 32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16–20, 2016, 2016, pp. 1–12.

[11] F. Tang, Q. Liu, H. Zhu, E. Chen, F. Zhu, Diversified social influence maximization, in: ASONAM, 2014, pp. 455–459.

[12] W. Chen, T. Lin, C. Yang, Efficient topic-aware influence maximization using preprocessing, 2014, CoRR, abs/1403.0057.

[13] P.M. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 26–29, 2001, 2001, pp. 57–66.

[14] J. Li, C. Liu, J.X. Yu, Y. Chen, T.K. Sellis, J.S. Culpepper, Personalized influential topic search via social network summarization, IEEE Trans. Knowl. Data Eng. 28 (7) (2016) 1820–1834.

[15] Y. Li, D. Zhang, K. Tan, Real-time targeted influence maximization for online advertisements, Proc. VLDB Endow. 8 (10) (2015) 1070–1081.

[16] Y. Li, J. Fan, D. Zhang, K.-L. Tan, Discovering your selling points: Personalized social influential tags exploration, in: Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17, ACM, New York, NY, USA, 2017, pp. 619–634.

[17] Y. Wang, G. Cong, G. Song, K. Xie, Community-based greedy algorithm for mining top-K influential nodes in mobile social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, ACM, New York, NY, USA, 2010, pp. 1039–1048.

[18] J. Li, X. Wang, K. Deng, X. Yang, T. Sellis, J.X. Yu, Most influential community search over large social networks, in: ICDE, April 2017, pp. 871–882.

[19] M. Wang, H. Li, J. Cui, K. Deng, S.S. Bhowmick, Z. Dong, PINOCCHIO: Probabilistic influence-based location selection over moving objects, IEEE Trans. Knowl. Data Eng. 28 (11) (2016) 3068–3082.

[20] L. Guo, D. Zhang, G. Cong, W. Wu, K.L. Tan, Influence maximization in trajectory databases, IEEE Trans. Knowl. Data Eng. 29 (3) (2017) 627–641.

[21] R. Li, J.X. Yu, R. Mao, Efficient core maintenance in large dynamic graphs, IEEE Trans. Knowl. Data Eng. 26 (10) (2014) 2453–2465.

[22] X. Huang, H. Cheng, L. Qin, W. Tian, J.X. Yu, Querying k-truss community in large and dynamic graphs, in: SIGMOD, 2014, pp. 1311–1322.

[23] C. Bron, J. Kerbosch, Finding all cliques of an undirected graph (algorithm 457), Commun. ACM 16 (9) (1973) 575–576.

[24] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (026113) (2004).

[25] M. Girvan, M. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 7821–7826.

[26] J. Ruan, W. Zhang, An efficient spectral algorithm for network community discovery and its applications to biological and social networks, in: IEEE ICDM, 2007, pp. 643–648.

[27] S. White, P. Smyth, A spectral clustering approach to finding communities in graph, in: SIAM- SDM, 2005, pp. 274–285.

[28] V. Satuluri, S. Parthasarathy, Scalable graph clustering using stochastic flows: applications to community discovery, in: ACM SIGKDD, 2009, pp. 737–746.
[29] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. 76 (3) (2007).
[30] M. Wang, C. Wang, J.X. Yu, J. Zhang, Community detection in social networks: An in-depth benchmarking study with a procedure-oriented framework, Proc. VLDB Endow. 8 (10) (2015) 998–1009.
[31] I.X. Leung, P. Hui, P. Lio, J. Crowcroft, Towards real-time community detection in large networks, Phys. Rev. 79 (6) (2009).
[32] G. Qi, C.C. Aggarwal, T.S. Huang, Community detection with edge content in social media networks, in: ICDE, 2012, pp. 534–545.
[33] E. Gregori, L. Lenzini, S. Mainardi, Parallel (k)-clique community detection on large-scale networks, IEEE Trans. Parallel Distrib. Syst. 24 (8) (2013) 1651–1660.
[34] J. Leskovec, K.J. Lang, M.W. Mahoney, Empirical comparison of algorithms for network community detection, in: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26–30, 2010, 2010, pp. 631–640.
[35] M.R. Vieira, H.L. Razente, M.C.N. Barioni, M. Hadjieleftheriou, D. Srivastava, C.T. Jr., V.J. Tsotras, DivDB: A system for diversifying query results, Proc. VLDB Endow. 4 (12) (2011) 1395–1398.
[36] M.R. Vieira, H.L. Razente, M.C.N. Barioni, M. Hadjieleftheriou, D. Srivastava, C.T. Jr., V.J. Tsotras, On query result diversification, in: ICDE, 2011, pp. 1163–1174.
[37] F. Zhao, X. Zhang, A.K.H. Tung, G. Chen, BROAD: diversified keyword search in databases, Proc. VLDB Endow. 4 (12) (2011) 1355–1358.
[38] B. Hu, Y. Zhang, W. Chen, G. Wang, Q. Yang, Characterizing search intent diversity into click models, in: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28–April 1, 2011, 2011, pp. 17–26.
[39] M.J. Welch, J. Cho, C. Olston, Search result diversity for informational queries, in: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28–April 1, 2011, 2011, pp. 237–246.
[40] V. Dang, W.B. Croft, Diversity by proportionality: an election-based approach to search result diversification, in: The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2012, 2012, pp. 65–74.
[41] X. Wang, Z. Dou, T. Sakai, J. Wen, Evaluating search result diversity using intent hierarchies, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, 2016, pp. 415–424.
[42] M. Drosou, E. Pitoura, Search result diversification, SIGMOD Rec. 39 (1) (2010) 41–47.
[43] B. Liu, G. Cong, Y. Zeng, D. Xu, Y.M. Chee, Influence spreading path and its application to the time constrained social influence maximization problem and beyond, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1904–1917.
[44] H. Lin, J.A. Bilmes, A class of submodular functions for document summarization, in: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19–24 June, 2011, Portland, Oregon, USA, 2011, pp. 510–520.
[45] K. Xie, K. Deng, S. Shang, X. Zhou, K. Zheng, Finding alternative shortest paths in spatial networks, ACM Trans. Database Syst. 37 (4) (2012) 29:1–29:31.
[46] D. Zhang, D. Yang, Y. Wang, K.-L. Tan, J. Cao, H.T. Shen, Distributed shortest path query processing on dynamic road networks, VLDB J. 26 (3) (2017) 399–419.
[47] M. Thorup, Integer priority queues with decrease key in constant time and the single source shortest paths problem, J. Comput. System Sci. 69 (3) (2004) 330–353.
[48] V.D. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of community hierarchies in large networks, 2008, CoRR, abs/0803.0476.