International Journal of Hybrid Information Technology
Vol.8, No.2 (2015), pp.409-422
http://dx.doi.org/10.14257/ijhit.2015.8.2.38

Identifying Topic-Sensitive Influential Spreaders in Social Networks

Donghao Zhou[1,2], Wenbao Han[2,3] and Yongjun Wang[1]

[1]*College of Computers, National University of Defense Technology, Changsha, China*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi, China*
[3]*College of Cyber Space Security, Information Engineering University, Zhengzhou, China*
dhzhou2084@163.com

Abstract

Identifying influential spreaders is an important issue in understanding the dynamics of information diffusion in social networks. It is to find a small subset of nodes, which can spread the information or influence to the largest number of nodes. The conventional approaches consider information diffusion through the network in a coarse-grained manner, without taking into account the topical features of information content and users. However, for messages with different topics, the target influential spreaders may vary largely.

In this paper, we propose to harness historical propagation data to learn the information diffusion probabilities on topic-level, based on which we use a greedy algorithm to iteratively select a set of influential nodes for a given topic. Specially, we design a three-stage algorithm named TopicRank to mine the most influential spreaders with respect to a specific topic. Given observed propagation data, we first use Latent Dirichlet Allocation (LDA) model to learn a topic mixture for each propagation message. Then, the topic-level diffusion probability of an edge is computed by exploiting the propagation actions occurred to it and the topic distribution of these propagation messages. Last, based on the learned topic-level diffusion probabilities, we apply optimized greedy algorithm CLEF to identify influential nodes with respect to a specific topic. Experimental results show that our method significantly outperforms state-of-the-art methods when used for topic-sensitive information spread maximization.

Keywords: *influential spreaders; information diffusion; information spread maximization; topic-sensitive; greedy algorithm*

1. Introduction

Diffusions of information and influence in social networks have received tremendous attention in the past few years. One of the key issues in this area is how to identify a small subset of influential spreaders, which can spread the information or influence to the largest number of nodes in social networks. The resolution of this issue can be helpful to many applications such as finding social leaders [1, 2], designing viral marketing strategies [3, 4] and searching domain experts [5].

This problem was first formalized by Kempe *et al.*, [6] as influence maximization problem, which is proved NP-hard. Although the problem is NP-hard, one can use a simple greedy algorithm to approximate the optimal solution with a theoretical guarantee of $(1-1/e)$. Following work [7] focused on reducing the computation complexity of the greedy algorithm

ISSN: 1738-9968 IJHIT

and applying it to larger scale social networks. Some other works [11, 12] proposed to identify influential nodes by taking advantage of the local or global structure of the network.

All these works, however, didn't take into consideration the topics of information content and nodes' preference. That is, they view a node's influence a constant, even for totally different topics. Obviously, this is inaccurate in real world, where a person, for example, may be very influential when talking about topics like "political affairs" but completely unknown in area of "sports". In other words, there is seldom all-round talent who is expert in all fields.

Besides, a key factor in the process of identifying influential spreaders is how to assign a diffusion probability for each edge. Traditional approaches often empirically choose a constant or draw values uniformly from a small set of constants for the influence probabilities, which will inevitably lead to inaccurate results. Only recently, researchers began to use real data like historical propagation traces to learn influence probabilities in social networks [13].

In this paper, we aim to harness historical propagation data, *i.e.*, the trace of information spreading across a network, to identify topic-sensitive influential nodes in social networks. That is, for a specific topic, finding a small set of nodes that can spread the information to the largest range of the network. In particular, to account for the topical features of information content and users in a social network, we design a three-stage algorithm, which we refer to as TopicRank, to rank the influential nodes on topic-level. The details of the algorithm are as follows:

(1) The first stage is topic distillation. We aggregate all propagation messages as a corpus and treat each message a mixture of various topics. We use Latent Dirichlet Allocation (LDA) [14] model to learn a topic distribution for each propagation message. The results of this step serve as a basis for the next stages.

(2) The second stage is topic-level diffusion probability computing. From the perspective of an edge, the topic-level diffusion probability attached to it is reflected by the number of information propagations that occurred to it and the information's topic distribution. We view the propagation actions as *implicit voting* on diffusion probability of edges. We devise a voting algorithm to learn the topic-level diffusion probability for edges in social networks.

(3) The last stage is ranking. We design a scheme to apply the topic-level diffusion probabilities for topic-sensitive influential nodes identifying, where a greedy algorithm is exploited to iteratively find the top-*k* influential spreaders in the diffusion network for a given topic.

The proposed approach is a data-driven and fine-grained approach, which can yield more accurate outcomes for topic-sensitive information diffusion maximization in real-world scenarios.

The remainder of this paper is organized as follows. Section 2 briefly surveys related work. Section 3 describes the data-driven approach used for learning the topic-level diffusion probability of edges. Section 4 elaborates the topic-sensitive influential nodes identifying algorithm TopicRank. We proceed by describing experimental evaluation in Section 5 and conclude in Section 6.

2. Related Work

Much effort has been made for social influence analysis and influential spreaders identifying and a large number of work has been done. Singla *et al.*, [15] proposed a method to qualitatively measure the existence of influence. Goyal *et al.*, [16] proposed

models and algorithms to learn influence probabilities in social networks using historical propagation data. As for influence maximization, Kempe *et al.*, [6] first formalized the problem and proposed a simple greedy algorithm to solve it. After that, Leskovec *et al.*, [8] exploited the submodularity property of influence function to propose an efficient greedy algorithm called CELF, which can effectively reduce the computation time of greedy algorithm. Kimura and Saito [9] considered shortest diffusion path to reduce the number of evaluations in Monte Carlo simulation process of the influence spread. Chen *et al.*, [7] instead considered Maximum influence Paths (MIP) to reduce the computation time under the IC model. Chen *et al.*, [10] also proposed a scalable heuristic called LDAG for the linear threshold model (LT model). Goyal *et al.*, [13] introduced a new model called credit distribution, that directly leverages available propagation traces to learn how influence flows in the network and uses this to estimate expected influence spread. All of these models didn't account for the topical features of information and nodes.

Some other researchers take into consideration the content and topics of information and proposed some topic specific models or algorithms. Topical Affinity Propagation (TAP) [17] models the topic-level social influence on large networks and demonstrates that different topics actually lead to different influence results. Pal *et al.*, [18] extracted multidimensional nodal and topical features to identify topical authorities in microblog network. Barbieri *et al.*, [19] studied social influence from a topic modeling perspective and introduced novel topic-aware influence-driven propagation models, and they claimed that the fine-grained model can be more accurate in describing real-world cascades than the standard propagation models. Accounting for users' different preferences on topics, Zhou *et al.*, [20] proposed a new GAUP algorithm to mine top-k influential nodes in social networks based on user preferences. GAUP is in essence an extension of greedy algorithm.

3. Computing topic-level Diffusion Probabilities

In this section, we first describe the information propagation data that we utilize. Then, we propose to use topic model like LDA to mine the topic distribution of the information content. Last, we devise an algorithm which is referred to as voting algorithm, to effectively compute diffusion probabilities of edges on topic-level.

3.1 Propagation data

Microblog is a new form of online social media, in which each user can choose who she wants to follow to receive messages from. In microblog, like Twitter, Weibo (the most popular microblog in China), a user publishes messages, and her followers may forward or comment the messages that they are interested. Such behaviors form information propagations, which is the main data source we consider in this paper. Formally, the microblog social network can be represented as a directed graph $G = (V, E)$ where V represents node set, and E represents edge set. For a piece of information c (a message, in other words), the propagation traces of c is called information cascade. A cascade is defined as $c := \{(u, v, t_u^c, t_v^c), \ldots, (w, z, t_w^c, t_z^c)\}$, where (u, v, t_u^c, t_v^c) means node u spreads information c to node v, and their infected time is t_u^c and t_v^c, respectively. We call $a_{u,v}^c := (u, v, t_u^c, t_v^c)$ a propagation action, which satisfies

$t_u^c \le t_v^c$. Note that, for each propagation action $a_{u,v}^c$, it corresponds to a specific edge $e_{u,v}$ and a time lag $\Delta t_{u,v}^c = t_v^c - t_u^c$. So, $a_{u,v}^c$ can also be written in the form of $a_{u,v}^c = (c, e_{u,v}, \Delta t_{u,v}^c)$. A cascade may have one or more propagation actions that are related to a specific message. All the observations form cascade set C.

All the propagation actions that correspond to $e_{i,j}$ are represented as $A_{i,j} = \{a_{i,j}^c \mid c \in C, \ and \ \ a_{i,j}^c \ exists \}$. From the perspective of edges, $A_{i,j}$ is the action set attached to edge $e_{i,j}$, and all propagation actions can be categorized according to N different edges. All the propagation actions in the data set is denoted as $\mathbf{A}$.

3.2. Topic Distillation

We view each of the propagation messages a mixture of various topics. The goal of topic distillation is to automatically identify the topic mixture of the messages. For this purpose, we choose topic model LDA, which is an unsupervised machine learning techniques to identify latent topics from large corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Figure 1 shows the graphical model of LDA.

In the graphical notation, shaded variable w indicates observed word tokens in a document, and unshaded variables φ, θ and z indicate latent (unobserved) variables. To generate a document d with N_d words, we first draw a topic distribution θ for it, which is governed by hyper parameter α. Then, for each word in d, we sample a topic z from θ, and for each topic z, we sample a word distribution φ for it. The word distribution φ is governed by hyper parameter β. Last, target word w is sampled according to topic z and its word distribution φ. Such process is repeated N_d times until all words have been generated for document d.

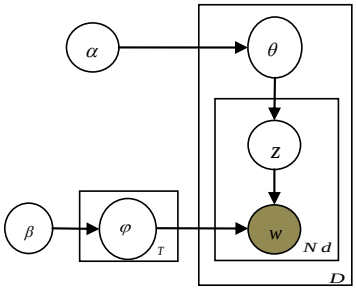


Figure 1. Graphical Model for Latent Dirichlet Allocation

In LDA, the variables θ and φ are the two sets of latent variables that we would like to infer, where θ represents the topic distribution of each document and φ indicates the word distribution for each topic. Here, we treat each piece of information in the propagation data as a single document, and all documents form the corpus.

Formally, we assume that there are K topics in the corpus, each represented by a word distribution. For each message c, it can be viewed as a mixture of topics, which is represented as a K-dimensional vector $\theta_c = (\theta_c^{(1)}, \theta_c^{(2)}, ..., \theta_c^{(K)})$ which satisfies $\sum_{i=1}^{i=K} \theta_c^{(i)} = 1$, where $\theta_c^{(i)}$ is the i th component of θ_c.

We set the number of topics to 100 and run 1000 iterations of Gibbs sampling using the GibbsLDA++ toolkit [1]. Finally, we get the target doc-topic matrix $C^{D\times K}$, where D indicates the number of documents (or messages), and K indicates the number of topics. The element $C_{i,j}^{D\times K}$ in $C^{D\times K}$ represents how many words in the i th document are assigned the j th topic. After normalization of each row vector in $C^{D\times K}$, one can get the topic distribution of each document . For a message c, that is θ_c.

3.3. Topic-level Diffusion Probability Computing

In this subsection, we will leverage propagation actions $\mathbf{A}$ and results from section 3.2 to learn topic-level diffusion probability of each edge. First, we introduce some useful notations. Let $p_{u,v}$ denote the diffusion probability of edge $e_{u,v}$. Assuming there are K topics, then on topic-level, $p_{u,v}$ can be represented as a K-dimensional vector $p_{u,v} = (p_{u,v}^{(1)}, p_{u,v}^{(2)}, ..., p_{u,v}^{(K)})$, where $p_{u,v}^{(i)}$ indicates the diffusion probability with respect to the i th topic.

Our approach is based on the following intuitions:

(1) The number of propagation actions executed by node u to node v reflect the diffusion probability of edge $e_{u,v}$. That is, the more times node u spreads messages to node v, the larger the diffusion probability of $e_{u,v}$.

(2) The topics of propagation messages reflected the diffusion probability on topic-level between two nodes. That is, for a specific edge $e_{u,v}$ and a given topic T, the more propagation actions occurred to $e_{u,v}$, and the higher the weights of topic T in topic distribution of the propagation messages, the larger the diffusion probability $p_{u,v}^{(T)}$.

In this way, a propagation action can be viewed as a vote, and the diffusion probability $p_{u,v}$ can be viewed as candidate. With this concept in mind, we can leverage hundreds of thousands of users' implicit voting on edges' diffusion probabilities. In this regard, our approach is a data-driven approach and is in accordance with the concept of Web 2.0 [2].

Now, we proceed by describing the voting process in details. For each edge and each topic, the topic-level diffusion probability is initialized as $p_{i,j}^{(T)} = 1/K$. If there exists a propagation action $a_{u,v}^{c}$, and the topic distribution of message c is θ_c, then $p_{i,j}^{(T)}$ is updated as

$$p_{i,j}^{(T)} = p_{i,j}^{(T)} + \theta_c^{(T)}. \quad (3.1)$$

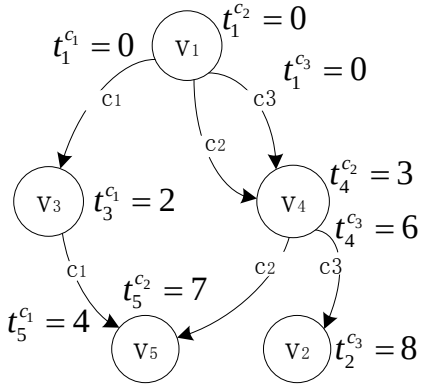
[1] http://sourceforge.net/projects/gibbslda/
[2] http://en.wikipedia.org/wiki/Web_2

By traversing all the propagation actions in $A_{i,j}$, and updating the votes on $p_{i,j}^{(T)}$, we finally get the total votes on topic-level diffusion probability for edge $e_{i,j}$.

Note that, in the naive voting algorithm (3.1), we ignored the time factor $\Delta t_{u,v}^{c}$ in propagation action $a_{u,v}^{c} = (c, e_{u,v}, \Delta t_{u,v}^{c})$. Previous work [16] shows that social influence decays over time in an exponential fashion, so propagation actions with different time delays $\Delta t_{u,v}^{c}$ should have different weights in the voting algorithm. Motivated by these ideas, we improve the updating process and rewrite it as

$$p_{i,j}^{(T)} = p_{i,j}^{(T)} + \theta_{c}^{(T)} \cdot \exp\{-\frac{t_j^c - t_i^c}{\tau_{i,j}}\}, \tag{3.2}$$

where $\tau_{i,j}$ is the mean life time. It corresponds to the expected time delay between node i performing an action and node j performing the same action. Equation (3.2) captures temporal feature of propagation data.



(a)

message	Topic distribution		
	health	technology	sport
c1	0.5	0.1	0.4
c2	0.1	0.8	0.1
c3	0.3	0.6	0.1

(b)

Figure 2. Information Propagation Graph. (a)Information Propagation Traces in a Social Network; and (b) Topic Distributions of Messages Over 3 Topics

Figure 2 shows an example of information propagation graph. There are three messages spread in the social network, which form three information cascades c_1, c_2 and c_3, respectively. From figure 2(a), we see two propagation actions that are attached to edge $e_{1,4}$, and one for each of others. Figure 2(b) illustrates the topic distributions of three messages over three topics (we set $K = 3$).

Then, according to (3.2), for topic "health", the diffusion probability between node 1 and node 4 is computed as

$$\begin{aligned} p_{1,4}^{(health)} &= 1/K + \theta_{c2}^{(health)} \times e^{-(t_4^{c2} - t_1^{c2})/\tau_{1,4}} + \theta_{c3}^{(health)} \times e^{-(t_4^{c3} - t_1^{c3})/\tau_{1,4}} \\ &= 1/3 + 0.1 \times e^{-3/\tau_{1,4}} + 0.3 \times e^{-6/\tau_{1,4}} \end{aligned}. \tag{3.3}$$

The last step is normalization. For each user j and a specific topic T, all the diffusion probabilities from its in-bound neighbors are normalized as

$$p_{i,j}^{(T)} = p_{i,j}^{(T)} / \sum_{i \in N(i)} p_{i,j}^{(T)}, \quad (3.4)$$

where $N(i)$ denotes the in-bound neighbors of node j. After normalization, the total diffusion probability from a node's in-bound neighbors equals to 1.

The whole algorithm is summarized in Algorithm 1.

Algorithm 1: Topic-level diffusion probability learning

Input: graph $G = (V, E)$, $A = \{A_{i,j} \mid (i,j) \in E\}, \{\theta_c\}_{c \in C}$, K

Output: $\{p_{i,j}^{(T)}\}_{(i,j) \in E, T \in [1:K]}$

initialization: $\{p_{i,j}^{(T)} = 1/K\}_{(i,j) \in E, T \in [1:K]}$
for each $cc \in C$ **do**
  for $a_{i,j}^{cc} \in cc$ **do**
    for $T = 1$ to K **do**
      $p_{i,j}^{(T)} = p_{i,j}^{(T)} + \theta_{cc}^{(T)} exp\{-\frac{t_j^{cc} - t_i^{cc}}{\tau_{i,j}}\}$
    end for
  end for
end for
for each $(i,j) \in E$ **do**
  for $T = 1$ to K **do**
    $p_{i,j}^{(T)} = p_{i,j}^{(T)} / \sum_{i \in N(i)} p_{i,j}^{(T)}$
  end for
end for

Online updating procedure. The proposed voting algorithm 1 can also be extended to handle the propagation data in an online style. For newly obtained information cascade, the online updating procedure sequentially updates the diffusion probabilities of edges that are related to the propagation data. Specially, for a newly coming cascade $nc = \{(u, v, t_u^{nc}, t_v^{nc}), ..., (w, z, t_w^{nc}, t_z^{nc})\}$, we first infer the topic distribution θ_{nc} of information nc based on the learned estimated parameters in LDA. Then, for each propagation action $a_{u,v}^{nc} \in nc$, $p_{u,v}$ is updated according to θ_{nc} and $a_{u,v}^{nc}$. Diffusion probabilities of edges that are not related to the propagation data are not to be updated, which makes the online updating procedure very effective in terms of computation.

4. Identifying Topic-Sensitive Influential Spreaders

In this section, we use the topic-level diffusion probabilities learned in section 3 to rank the influential nodes. We propose to use greedy hill-climbing algorithm to identify influential spreaders on topic-level in social networks.

4.1. Topic-level diffusion graph

Based on the topic-level diffusion probabilities, we are able to build a multi-level information diffusion graph with each level corresponding to a specific topic, as shown in figure 3.

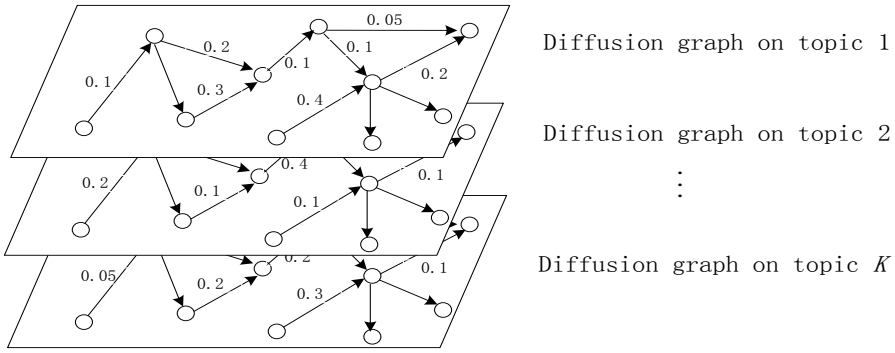


Figure 3. Topic-level Information Diffusion Graphs

For a specific topic T, the information diffusion graph with respect to it is a weighted graph. We denoted it as $G = (V, E, P^{(T)})$, where $V = \{v_i\}$, $E = \{e_{i,j}\}$, $P^{(T)} = \{P_{i,j}^{(T)} \mid e_{i,j} \in E\}$ denotes vertex, edges and topic specific diffusion probabilities. In this way, we can model the target social network in a fine-grained manner, that is, for K topics, we have K different diffusion graphs. The following proposed approach is based on this topic-level information diffusion graphs.

4.2. Greedy Algorithm

Problem definition. Let notation $\sigma_T(S)$ denotes the spread of information if we choose nodes set S to publish the information with topic label T. The problem of identifying topic-sensitive influential nodes can be formulated as: for a given topic label T and a number k, finding a seed set S, $|S| = k$, which can maximize $\sigma_T(S)$.

This problem has been proved to be NP-hard [6]. It is not realistic to find an optimal solution. However, Kempe et al [6] proves that the influence function under IC model and LT model is monotone and submodular. A function $\sigma(\cdot)$ is submodular if it satisfies a natural ‘‘diminishing returns’’ property, i.e., if $S \subseteq T$, then we have $\sigma(S \cup \{v\}) - \sigma(S) \geq \sigma(T \cup \{v\}) - \sigma(T)$. For submodular and monotone function $\sigma(\cdot)$ with $\sigma(\varnothing) = 0$, the problem of finding a set S of size k that maximizes $\sigma(S)$ can be approximated by a simple greedy algorithm which is shown in Algorithm 2. The idea of the greedy algorithm is to run for k rounds, where each round finds the vertex that will maximize the incremental influence spread in the round.

To evaluate the expected spread range, we use Monte Carlo simulations under Independent Cascade (IC) model, to mimic information spreading across the social network. In this process, two factors are very important to the final results. The first is activation probabilities assigned to edges and the second is the evaluation of influence spread. For the first factor, in the traditional IC model, the probability of activation is uniform to all edges. We extend the

IC model and propose to use the topic specific diffusion probability $\{P_{i,j}^{(T)}\}_{e_{i,j}\in E}$ as activation probability, which captures the preferences of users on different topics.

Algorithm 2: Traditional greedy algorithm $GA(k,\sigma(\cdot))$

Input: graph $G=(V,E)$, seed size k, influence function $\sigma(\cdot)$

Output: seed set S, s.t. $|S|=k$

1: Initialization: $S=\varnothing$
2: **for** i=1 to k **do**
3: $v=\arg\max_{u\in V\setminus S}(\sigma(S\cup\{u\})-\sigma(S))$
4: $S=S\cup\{v\}$
5: **end for**

As for the second factor, the traditional influence maximization algorithm often views the number of finally activated nodes as expected value of the influence function. Here, taking into account the topic preference of users, we propose to use topic specific weights of the finally activated nodes as the topic specific information spread (*TSIS*), that is

$$TSIS_T(S)=\sum_{u\in S}\theta_u^{(T)}=\sum_{u\in S}\sum_{v\in N_{out}(u)}p_{u,v}^{(T)}, \tag{4.1}$$

where $\theta_u^{(T)}$ means the weight of topic T in node u's preference. It can be calculated by summing all its out-bound diffusion probabilities on topic T.

We describe the details of proposed greedy algorithm in Algorithm 3. First we compute the topic-level diffusion probability $P_{i,j}^{(T)}$ for edges according Algorithm 1, and user's topic preference $\theta_u^{(T)}$, which are used in the Monte Carlo simulations. In the greedy hill-climbing process, we iteratively run k rounds, and in each round we choose the seed which can make the largest marginal gain for influence spread with respect to topic T. We keep a record of a node's marginal gain in tuple-2 data structure $<node, gain>$. All the records are stored in queue Q and sorted by the marginal gain value in descending order. In algorithm 3, cis denotes current information spread of seed set S. In round i, we first check whether the marginal gain of the first-position node in Q has been updated in current round. If the answer is yes, it is to choose as target seed in this round. If not, then the marginal gain of it will be updated and inserted into queue Q again (maybe not at the first position). The procedure is repeated until we find the target node (line 16-26).

Note that, from line 13-28, we implement the optimized CLEF [8] algorithm. By exploiting submodularity property of influence function, in many cases, we don't need to re-compute all the nodes' marginal gain to find the best node. For example, if the top element still stays the top element even after recomputation, then there is no need to reevaluate other node's marginal gain. Leskovec et al [8] claimed that CLEF can speed up the greedy algorithm by 700 times.

Algorithm 3: Topic-sensitive influential nodes identifying

Input: graph $G = (V, E, P^{(T)})$, topic T, seed size k

Output: seed set S, s.t. $|S| = k$

Initialization: $S = \varnothing$

Pre-computing of topic-level diffusion probability for edges $\{P_{i,j}^{(T)} \mid e_{i,j} \in E\}$

Pre-computing of users' topic preference $\theta_u^{(T)} = \sum_{v \in N_{out}(u)} p_{u,v}^{(T)}$

//*Select the first seed*

for each vertex $u \in V$ **do**

  computation of node u's influence spread $TSIS_T(u)$

  $cn.node = u, cn.gain = TSIS_T(u)$

  insert cn to Q

end for

$cn = pop(Q)$

$S = \{cn.node\}, cis = cn.gain$

//*Select the next* $k-1$ *seeds*

for $i = 2$ to k **do**

  $VS = \varnothing$ /*examined nodes set in current round*/

  while True **do**

    $cn = pop(Q)$, $v = cn.node$

    if $v \in VS$ **then**

      $S = S \cup \{v\}$

      $cis = cis + v.gain$

      break

    else

      computation of $TSIS_T(S \cup \{v\})$

      node v's marginal gain: $TSIS_T(S \cup \{v\}) - cis$

      $cn.node = v, cn.gain = TSIS_T(S \cup \{v\}) - cis$

      insert cn to Q

      $VS = VS \cup \{v\}$

  end while

end for

4. Experimental Evaluation

In this section, we evaluate the performance of our proposed approach TopicRank by comparing to three baselines, including traditional greedy algorithm, PageRank and Topic-sensitive PageRank. We conduct a series of experiments on real-world data set to answer the question of, given a specific topic, how well does our approach perform in terms of information spread maximization compared with the baseline algorithms.

4.1. Experiment Setup

Dataset. The dataset used is crawled from Sina Weibo[1], China's the most popular microblog site. We crawled a subset of users and their posts from March of 2013 to March of 2014. The network has 61,605 nodes and 1,631,228 edges. We extract 10,600 cascades by tracing the spread of special hashtags (in form of #meme name#). The average size of these cascades is 32.4. Totally 35,065 unique users participate at least one cascade.

Baseline. Three related algorithms are used as baseline, *i.e.*, traditional greedy algorithm (GA), PageRank (PR) and Topic-sensitive PageRank (TSPR).

- Greedy algorithm (GA). The idea of greedy algorithm is to run for *K* rounds, where each round finds the vertex that will maximize the incremental influence spread in the round.
- PageRank (PR). It works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.
- Topic-sensitive PageRank (TSPR). The key advantage of topic-sensitive PageRank is that it can bias the computation to increase the effect of certain topics of pages by using a nonuniform personalization vector. But the transmission matrix it used is the same as that used in PageRank. Here, we set the teleportation parameter $\alpha = 0.15$, and the personalization vector $\overline{p} = \frac{1}{Z}[\theta_1^{(T)}, \theta_2^{(T)}, \cdots, \theta_N^{(T)}]$, where Z is a normalizing factor.

Metrics. For the problem of influence spread, we choose topic specific information spread $TSIS_T(S)$ as metric, which can effectively measure the maximum influence of the seed set with respect to a given topic.

4.2. Experimental Results

Here, we consider 3 totally different topics in our experiments, which are "machine learning & data mining", "smart phone" and "health & sports". The topic of "machine learning & data mining" (ML&DM) is very "local" and niche, with only a few professional researchers and university students being interested in it. While topic "smart phone" attract more people compared to topic ML&DM, it is not as general as topic "health & sports". The three topics have different weights in microblog messages on the whole, with a descending order of "health & sports", "smart phone" and "ML&DM".

Figure 4 to 6 demonstrate the topic-sensitive information spread (TSIS) of various algorithms on three topics, respectively. We vary the seed set size from 5 to 60. From the figures, we can see that for each of the three topics, PageRank as the baseline performs worst. Greedy algorithm without considering topics performs a little better than PageRank. Two algorithms, TopicRank and Topci-sensitive PageRank, which take into account topics, are at the first and second place. This is because TopicRank and Topci-sensitive PageRank can find the most influential nodes with respect to the specific topic, while PageRank and traditional greedy algorithm can only find influential nodes independent of given topics.

In particular, for topic "ML&DM", TopicRank yields an information spread result of 282 when the seed set size increase to 60, which is about 50% higher than the runner-up algorithm

[1] http://weibo.com

TSPR, and 133% higher than greedy algorithm. Note that, for such a topic, even the best spread is smaller than 300, which verifies that "ML&DM" is truly a small community.

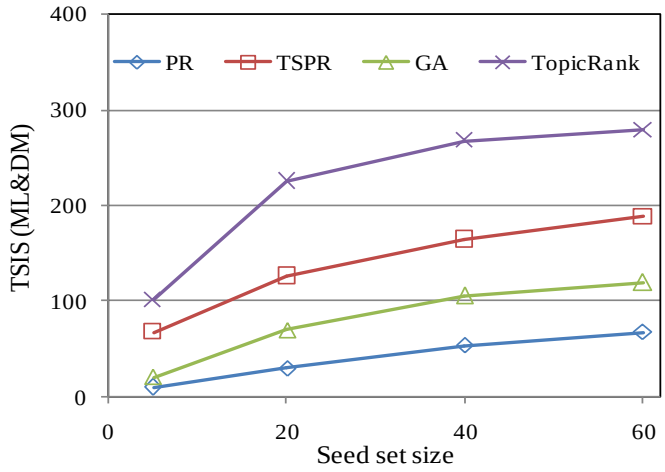


Figure 4. Information Spread with Respect to Topic "Machine Learning and Data Mining"

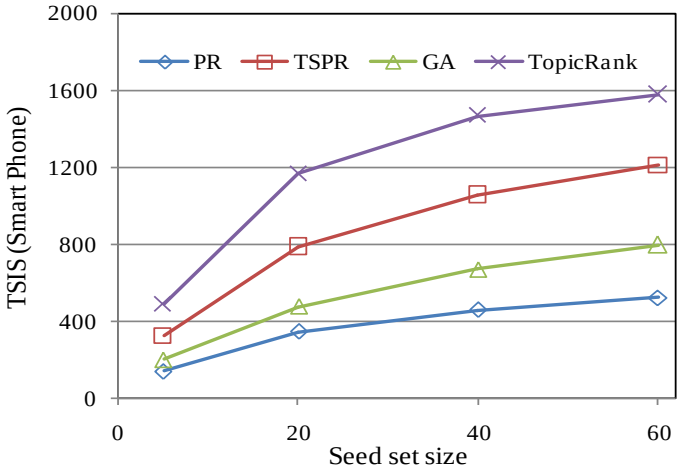


Figure 5. Information Spread with Respect to Topic "Smart Phone"

For topic "smart phone" and "health & sports", when seed set size is 60, TopicRank outperforms Topic-sensitive PageRank by about 30% and 19%, respectively. We infer that the advantage of TopicRank is broader for "local" and niche topics than general topics. That is to say, the more professional and "local" a topic is, the influential nodes with respect to it found by our proposed algorithm are more accurate and effective. Because the topic "health & sports" is very popular in social network like microblog, the distinctness of TopicRank for such topics is not as obvious as that for "ML&DM".

The results in figure 4 to 6 indicate that by considering the topical feature of information content, our proposed algorithm TopicRank can identify influential nodes with respect to a given topic more accurately, and thus can promote the topic-sensitive information spread more effectively.

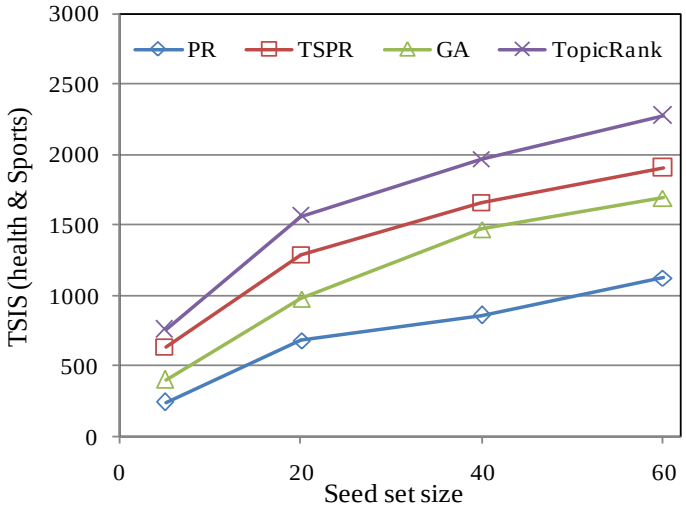


Figure 6. Information Spread with Respect to Topic "Health & Sports"

5. Conclusions

In this paper, we aim to exploit historical propagation data to identify influential nodes on topic-level. Our proposed approach TopicRank starts from the observation that for a given topic T , the diffusion probability of an edge is reflected by number of propagation actions occurred to the edge and the weights of topic T in topic distribution of the propagation messages. We first use LDA model to learn topic distributions for all the propagation messages. Then a novel voting algorithm is proposed to learn diffusion probabilities of edges on topic-level, by leveraging the propagation actions and topic distributions of messages. Last, for a given topic, we use an optimized greedy algorithm to iteratively find the top-*k* influential spreaders in the diffusion network. Experimental results show that the proposed approach outperforms state-of-the-art models in terms of topic-sensitive information spread.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No.61271252 and No.61202482), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No.20124307110014), and fund from the State Key Laboratory of Mathematical Engineering and Advanced Computing of China.

References

[1] L. Lü, Y. C. Zhang, H. C. Yeung and T. Zhou, "Leaders in social networks, the delicious case", PLoS ONE, vol. 6, 21202, **(2011)**.
[2] D. Chen, L. Lü, M. S. Shang, Y. Zhang and T. Zhou, "Identifying influential nodes in complex networks", Physica a: Statistical mechanics and its applications, vol. 391, no. 4, **(2012)**, pp. 1777-1787.
[3] P. Domingos and M. Richardson, "Mining the network value of customers", In Pro. of the Seventh ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, **(2001)**, pp. 57-66.
[4] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing", In Pro. of the Eighth ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, **(2002)**, pp. 61-70.

[5] J. Zhang, M. S. Ackerman and L. Adamic, "Expertise networks in online communities: structure and algorithms", In 16th International Conference of World Wide Web, Banff, Alberta, Canada, **(2007)**, pp. 221-230.
[6] D. Kempe, J. Kleinberg and E. Tardos, "Maximizing the spread of influence through a social network", In Proc. of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, USA, **(2003)**,pp. 137-146.
[7] W. Chen, C. Wang and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks", In Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York0, **(2010)**, pp. 1029-1038.
[8] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen and N. Glance, "Cost-effective Outbreak Detection in Networks", In Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, **(2007)**, pp. 420-429.
[9] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks", In Knowledge Discovery in Databases (PKDD 2006), Springer, Berlin Heidelberg, **(2006)**, pp. 259-271.
[10] W. Chen, Y. Yuan and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model", In 10th International Conference on Data Mining, IEEE, New York, **(2010)**, pp. 88-97.
[11] M. Kitsak, L. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. Stanley and H. Makse, "Identification of influential spreaders in complex networks", Nature Physics, vol. 6, no. 11, **(2010)**, pp. 888-893.
[12] B. Hou, Y. Yao and D. Liao, "Identifying all-around nodes for spreading dynamics in complex networks", Physica A: Statistical Mechanics and its Applications, vol. 391, no. 15, **(2012)**, pp. 4012-4017.
[13] A. Goyal, F. Bonchi and L. V. Lakshmanan, "A data-based approach to social influence maximization", Proceedings of the VLDB Endowment, vol. 5, no. 1, **(2011)**, pp. 73-84.
[14] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation", The Journal of Machine Learning Research, vol. 3, no. 1, **(2003)**, pp. 993-1022.
[15] P. Singla and M. Richardson, "Yes, there is a correlation: from social networks to personal behavior on the web", In Proc. of the 17th International Conference on World Wide Web, ACM, New York, (**2008**), pp. 655–664.
[16] A. Goyal, F. Bonchi and L. V. Lakshmanan, "Learning influence probabilities in social networks", In Proc. of the Third ACM International Conference on Web Search and Data Mining, ACM, New York, **(2010)**, pp. 241-250.
[17] J. Tang, J. Sun, C. Wang and Z. Yang, "Social influence analysis in large-scale networks", In Proc. of the 15th ACMSIGKDD International Conference On Knowledge Discovery and Data Mining, ACM, New York, **(2009)**, pp.807–816.
[18] A. Pal and S. Counts, "Identifying topical authorities in microblogs", In Proc. of the fourth ACM International Conference on Web Search and Data Mining, ACM, New York, **(2011)**, pp. 45-54.
[19] N. Barbieri, F. Bonchi and G. Manco, "Topic-aware social influence propagation models", Knowledge and information systems, vol. 37, no. 3, **(2013)**, pp. 555-584.
[20] J. Zhou, Y. Zhang and J. Cheng, "Preference-based mining of top-K influential nodes in social networks", Future Generation Computer Systems, vol. 31, **(2014)**, pp. 40-47.

