Real-time Topic-aware Influence Maximization Using Preprocessing

Wei Chen, Tian Lin, and Cheng Yang

Microsoft Research, weic@microsoft.com
Tsinghua University, lint10@mails.tsinghua.edu.cn
Tsinghua University, albertyang33@gmail.com

Abstract. Influence maximization is the task of finding a set of seed nodes in a social network such that the influence spread of these seed nodes based on certain influence diffusion model is maximized. Topic-aware influence diffusion models have been recently proposed to address the issue that influence between a pair of users are often topic-dependent and information, ideas, innovations etc. being propagated in networks are typically mixtures of topics. In this paper, we focus on the topic-aware influence maximization task. In particular, we study preprocessing methods to avoid redoing influence maximization for each mixture from scratch. We explore two preprocessing algorithms with theoretical justifications. Our empirical results on data obtained in a couple of existing studies demonstrate that one of our algorithms stands out as a strong candidate providing microsecond online response time and competitive influence spread, with reasonable preprocessing effort.

Keywords: influence maximization, topic-aware influence modeling, information diffusion

1 Introduction

In a social network, information, ideas, rumors, and innovations can be propagated to a large number of people because of the social influence between the connected peers in the network. *Influence maximization* is the task of finding a set of *seed nodes* in a social network such that the influence propagated from the seed nodes can reach the largest number of people in the network. More technically, a social network is modeled as a graph with nodes representing individuals and directed edges representing influence relationships. The network is associated with a stochastic diffusion model (such as independent cascade model and linear threshold model [14]) characterizing the influence propagation dynamics starting from the seed nodes. Influence maximization is to find a set of k seed nodes in the network such that the *influence spread*, defined as the expected number of nodes influenced (or activated) through influence diffusion starting from the seed nodes, is maximized [14,6].

Influence maximization has a wide range of applications including viral marketing [9,18,14], information monitoring and outbreak detection [15], competitive viral marketing and rumor control [5,13], or even text summarization [22] (by modeling a word influence network). As a result, influence maximization has been extensively studied in the past decade. Research directions include improvements in the efficiency and scalability of influence maximization algorithms [8,21,12], extensions to other diffusion models and optimization problems [5,3,13], and influence model learning from real-world data [19,20,10].

Most of these works treat diffusions of all information, rumors, ideas, etc. (collectively referred as *items* in this paper) as following the same model with a single set of parameters. In reality, however, influence between a pair of friends may differ depending on the topic. For example, one may be more influential to the other on high-tech gadgets, while the other is more influential on fashion topics, or one researcher is more influential on data mining topics to her peers but less influential on algorithm and theory topics. Recently, Barbieri et al. [2] propose the topic-aware independent cascade (TIC) and linear threshold (TLT) models, in which a diffusion item is a mixture of topics and influence parameters for each item are also mixtures of parameters for individual topics. They provide learning methods to learn influence parameters in the topic-aware models from real-world data. Such topic-mixing models require new thinking in terms of the influence maximization task, which is what we address in this paper.

In this paper, we adopt the models proposed in [2] and study efficient topic-aware influence maximization schemes, i.e., finding a set of k seed nodes to trigger the information cascade whenever a diffusion item composed of multiple topics is given. It has a wide application in viral marketing for online scenarios, where the system should recommend candidate sets instantly to different queries. One can still apply topic-oblivious influence maximization algorithms in online processing of every diffusion item, but it may not be efficient when there are a large number of items with different topic mixtures or real-time responses are required. Thus, our focus is on how to utilize the preprocessing of individual topic influence so that when a diffusion item with certain topic mixture comes, the online processing of finding the seed set is fast. To do so, our first step is to collect two datasets in the past studies with available topic-aware influence analysis results on real networks and investigate their properties pertaining to our preprocessing purpose. Our data observation shows that in one network users and their relationships are largely separated by different topics while in the other network they have significant overlaps on different topics. Even with this difference, a common property we find is that in both datasets most top seeds for a topic mixture come from top seeds of the constituent topics, which matches our intuition that influential individuals for a mixed item are usually influential in at least one topic category.

Motivated by our findings from the data observation, we explore two preprocessing based algorithms (Section 3). The first algorithm, *Best Topic Selection* (BTS), minimizes online processing by simply using a seed set for one of the constituent topics. Even for such a simple algorithm, we are able to provide a theoretical approximation ratio (when a certain property holds), and thus BTS serves as a baseline for preprocessing algorithms. The second algorithm, *Marginal*

Influence Sort (MIS), further uses pre-computed marginal influence of seeds on each topic to avoid slow greedy computation. We provide a theoretical justification showing that MIS can be as good as the offline greedy algorithm when nodes are fully separated by topics.

We then conduct experimental evaluations of these algorithms and comparing them with both the greedy algorithm and a state-of-the-art heuristic algorithm PMIA [21], on the two datasets used in data analysis as well as a third dataset for testing scalability (Section 4). From our results, we see that MIS algorithm stands out as the best candidate for preprocessing based real-time influence maximization: it finishes online processing within a few microseconds and its influence spread either matches or is very close to that of the greedy algorithm. Full technical details including data analysis, proofs and experimental results are available in the technical report [7].

Our work, together with a recent independent work [1], is one of the first that study topic-aware influence maximization with focus on preprocessing. Comparing to [1], our contributions include: (a) we include data analysis on two real-world datasets with learned influence parameters, which shows different topical influence properties and motivates our algorithm design; (b) we provide theoretical justifications to our algorithms; (c) the use of marginal influence of seeds in individual topics in MIS is novel, and is complementary to the approach in [1]; (d) although MIS is simple, it achieves competitive influence spread within microseconds of online processing time satisfying real-time application requirement.

2 Preliminaries

In this section, we introduce the background and problem definition on the topic-aware influence diffusion models. We focus on the independent cascade model [14] for ease of presentation, but our results also hold for other models parameterized with edge parameters such as the linear threshold model [14].

Independent cascade model. We consider a social network as a directed graph $G = (V, E)$, where each node in V represents a user, and each edge in E represents the relationship between two users. For every edge $(u, v) \in E$, denote its *influence probability* as $p(u, v) \in [0, 1]$, and we assume $p(u, v) = 0$ for all $(u, v) \notin E$ or $u = v$. The *independent cascade (IC)* model, defined in [14], captures the stochastic process of contagion in discrete time. Initially at time step $t = 0$, a set of nodes $S \subseteq V$ called *seed nodes* are activated. At any time $t \geq 1$, if node u is activated at time $t - 1$, it has one chance of activating each of its inactive outgoing neighbor v with probability $p(u, v)$. A node stays active after it is activated. This process stops when no more nodes are activated.

We define *influence spread* of seed set S under influence probability function p, denoted $\sigma(S, p)$, as the expected number of active nodes after the diffusion process ends. As shown in [14], for any fixed p, $\sigma(S, p)$ is monotone (i.e., $\sigma(S, p) \leq \sigma(T, p)$ for any $S \subseteq T$) and submodular (i.e., $\sigma(S \cup \{v\}, p) - \sigma(S, p) \geq \sigma(T \cup \{v\}, p) - \sigma(T, p)$ for any $S \subseteq T$ and $v \in V$) in its seed set parameter. For two influence probability functions p and p' on graph $G = (V, E)$, we denote $p \leq p'$ if for any $(u, v) \in E$, $p(u, v) \leq p'(u, v)$. Another well-known fact is that $\sigma(S, p)$ is *monotone in* p (i.e. $\sigma(S, p) \leq \sigma(S, p')$ if $p \leq p'$ edge-wise).

Influence maximization. Given a graph $G = (V, E)$, an influence probability function p, and a budget k, *influence maximization* is the task of selecting at most k seed nodes in V such that the influence spread is maximized, i.e., finding the optimal seeds $S^* = S^*(k, p) \subseteq V$ such that $S^* = \mathrm{argmax}_{S \subseteq V, |S| \leq k}\, \sigma(S, p)$.

Kempe et al. [14] show that the influence maximization problem is NP-hard in both the IC and LT models, and they propose the following *greedy algorithm*. Given influence probability function p, the *marginal influence (MI)* of any node $v \in V$ under any seed set S is defined as $MI(v|S, p) = \sigma(S \cup \{v\}, p) - \sigma(S, p)$. The greedy algorithm selects k seeds in the following k iterations: (a) let $S_0 = \emptyset$; (b) for each iteration $j = 1, 2, \ldots, k$, find node $v_j = \mathrm{argmax}_{v \in V \setminus S_{j-1}}\, MI(v|S_{j-1}, p)$, and adds v_j into S_{j-1} to obtain S_j; (c) output seed set $S^g(k, p) = S_k$.

It is shown in [14] that the greedy algorithm selects a seed set $S^g(k, p)$ with approximation ratio $1 - \frac{1}{e} - \varepsilon$ for any small $\varepsilon > 0$ (i.e., $\sigma(S^g(k, p), p) \geq \left(1 - \frac{1}{e} - \varepsilon\right) \sigma(S^*, p)$), where ε accommodates the inaccuracy in Monte Carlo simulations to estimate the marginal influence.

Topic-aware independent cascade model and topic-aware influence maximization. Topic-aware independent cascade (TIC) model [2] is an extension of the IC model to incorporate topic mixtures in any diffusion item. Suppose there are d base topics, and we use set notation $[d] = \{1, 2, \cdots, d\}$ to denote topic $1, 2, \cdots, d$. We regard each diffusion item as a distribution of these topics. Thus, any item can be expressed as a vector $I = (\lambda_1, \lambda_2, \ldots, \lambda_d) \in [0, 1]^d$ where $\sum_{i \in [d]} \lambda_i = 1$. We also refer such a vector I as a *topic mixture*. Given a directed social graph $G = (V, E)$, influence probability on any topic $i \in [d]$ is $p_i : V \times V \to [0, 1]$, and we assume $p_i(u, v) = 0$ for all $(u, v) \notin E$ or $u = v$. In the TIC model, the influence probability function p for any diffusion item I is defined as $p(u, v) = \sum_{i \in [d]} \lambda_i p_i(u, v)$, for all $u, v \in V$ (or simply $p = \sum_{i \in [d]} \lambda_i p_i$). Then, the stochastic diffusion process and influence spread $\sigma(S, p)$ are exactly the same as defined in the IC model by using the influence probability p on edges.

Given a social graph G, base topics $[d]$, influence probability function p_i for each base topic i, a budget k and an item $I = (\lambda_1, \lambda_2, \ldots, \lambda_d)$, the *topic-aware influence maximization* is the task of finding optimal seeds $S^* = S^*(k, p) \subseteq V$ such that $S^* = \mathrm{argmax}_{S \subseteq V, |S| \leq k}\, \sigma(S, p)$, where $p = \sum_{i \in [d]} \lambda_i p_i$.

3 Preprocessing Based Algorithms

Topic-aware influence maximization can be solved by using existing influence maximization algorithms such as the ones in [14,21]: when a query on an item $I = (\lambda_1, \lambda_2, \cdots, \lambda_d)$ comes, the algorithm first computes the mixed influence probability function $p = \sum_j \lambda_j p_j$, and then applies existing algorithms using parameter p. This, however, means that for each topic mixture influence maximization has to be carried out from scratch. It may take from half a minute to several hours to find the seed sets in large-scale networks, which could be inefficient or impractical for online scenarios.

In this paper, we are able to obtain datasets from two prior studies, one is on social movie rating network Flixster [2] and the other is on academic collaboration network Arnetminer [20], to help design our algorithms. Due to the space limit, the full data analysis can be found in [7], and we briefly summarize two key observations we made as follows: (1) Topic separation in terms of influence probabilities is network dependent: In the Arnetminer network, topics are mostly separated among different edges and nodes in the network, while in the Flixster network there are significant overlaps on topics among nodes and edges; (2) Most seeds for topic mixtures come from the seeds of constituent topics, in both Arnetminer and Flixster networks. In this section, motivated by the above observations, we introduce two preprocessing based algorithms that cover different design choices.

3.1 Best Topic Selection (BTS) algorithm

Our first algorithm is to minimize online processing by simply selecting a seed set from one of the constituent topics that has the best influence spread in the topic mixture, and thus we call it *Best Topic Selection (BTS)* algorithm. Since the query of item $I = (\lambda_1, \lambda_2, \cdots, \lambda_d)$ may be arbitrary, our key idea is to apply a bucketing technique to establish landmarks for each topic in the preprocessing stage, and use properties of upper and lower landmarks to bound the error in the online stage, as we explain in more detail now.

Preprocess stage. Denote constant set $\Lambda = \{\lambda^c_0, \lambda^c_1, \cdots, \lambda^c_m\}$ as a set of *landmarks*, where $0 = \lambda^c_0 < \lambda^c_1 < \cdots < \lambda^c_m = 1$. For each $\lambda \in \Lambda$ and each topic $i \in [d]$, we pre-compute $S^g(k, \lambda p_i)$ and $\sigma(S^g(k, \lambda p_i), \lambda p_i)$ in the preprocessing stage, and store these values for online processing. In our experiments, we use uniformly selected landmarks because they are good enough for influence maximization and can adopt parallel optimization. More sophisticated landmark selection method may be applied, such as the machine learning based method in [1].

Online stage. We define two rounding notations that return one of the neighboring landmarks in $\Lambda = \{\lambda^c_0, \lambda^c_1, \cdots, \lambda^c_m\}$: given any $\lambda \in [0,1]$, let $\underline{\lambda} = \lambda^c_j$ such that $\lambda^c_j \le \lambda < \lambda^c_{j+1}$, and $\overline{\lambda} = \lambda^c_{j+1}$ such that $\lambda^c_j < \lambda \le \lambda^c_{j+1}$. Given $I = (\lambda_1, \lambda_2, \cdots, \lambda_d)$, let $D^+_I = \{i \in [d] \,|\, \lambda_i > 0\}$. With the pre-computed $S^g(k, \lambda p_i)$ and $\sigma(S^g(k, \lambda p_i), \lambda p_i)$ for every $\lambda \in \Lambda$ and every topic i, the BTS algorithm is given in Algorithm 1. The algorithm basically rounds down the mixing coefficient on every topic to $(\underline{\lambda}_1, \cdots, \underline{\lambda}_d)$, and then returns the seed set $S^g(k, \underline{\lambda}_{i'} p_{i'})$ that gives the largest influence spread at the round-down landmarks.

In this paper, BTS is used as a baseline for preprocessing based algorithms. Although BTS is rather simple, we show below that it could provide theoretical guarantee with a certain condition.

We say that $\sigma(S, p)$ is *c-sub-additive in p* for some constant c if for any $S \subseteq V$ with $|S| \le k$ and any $I = (\lambda_1, \ldots, \lambda_d)$, $\sigma(S, \sum_{i \in D^+_I} \lambda_i p_i) \le c \sum_{i \in D^+_I} \sigma(S, \lambda_i p_i)$.

Algorithm 1 Best Topic Selection (BTS) Algorithm

```
Require: $G = (V, E)$, $k$, $\{p_i \mid i \in [d]\}$, $I = (\lambda_1, \cdots, \lambda_d)$, $\Lambda$, $S^g(k, \lambda p_i)$ and
    $\sigma(S^g(k, \lambda p_i), \lambda p_i)$, $\forall \lambda \in \Lambda, \forall i \in [d]$.
1: $I' = (\underline{\lambda}_1, \cdots, \underline{\lambda}_d)$
2: $i' = \mathrm{argmax}_{i \in D_I^+} \sigma(S^g(k, \underline{\lambda}_i p_i), \underline{\lambda}_i p_i)$
3: return $S^g(k, \underline{\lambda}_{i'} p_{i'})$
```

The sub-additivity property above means that the influence spread of any seed set S in any topic mixture will not exceed constant times of the sum of the influence spread for each individual topic. It is easy to verify that, when each topic in the network does not interfere with each other, $\sigma(S, p)$ is 1-sub-additive. The counterexample we could find that violates the c-sub-additivity assumption is a tree structure where even layer edges are for one topic and odd layer edges are for another topic. Such structures are rather artificial, and we believe that for real networks the influence spread is c-sub-additive in p with a reasonably small c.

We define $\mu_{\max} = \max_{i \in [d], \lambda \in [0,1]} \frac{\sigma(S^g(k, \overline{\lambda} p_i), \overline{\lambda} p_i)}{\sigma(S^g(k, \underline{\lambda} p_i), \underline{\lambda} p_i)}$, which is a value controlled by preprocessing. A fine-grained landmark set Λ could make $\mu_{\max}$ close to 1. The following Theorem 1 guarantees the approximation ratio of Algorithm 1.

Theorem 1. *If the influence spread function $\sigma(S, p)$ is c-sub-additive in p, Algorithm 1 achieves $\frac{1 - e^{-1}}{c |D_I^+| \mu_{\max}}$ approximation ratio for item $I = (\lambda_1, \lambda_2, \cdots, \lambda_d)$.*

The approximation ratio given in the theorem is a conservative bound for the worst case (e.g., a common setting may be $c = 1.2$, $\mu_{\max} = 1.5$, $|D_I^+| = 2$). Tighter online bound in our experiment section based on [15] shows that Algorithm 1 performs much better than the worst case scenario.

3.2 Marginal Influence Sort (MIS) algorithm

Our second algorithm derives the seed set from constituent topics, and moreover it utilizes pre-computed marginal influence from different topics to select seeds. Our idea is partially motivated by our data observation, especially for the Arnetminer dataset, which shows that in some cases the network could be well separated among different topics. Intuitively, if nodes are separable among different topics, and each node v is only pertinent to one topic i, the marginal influence of v would not change much whether it is for a mixed item or the pure topic i, as formally characterized in the following. Given threshold $\theta \geq 0$, define node set $\nu_i(\theta) = \{v \in V \mid \sum_{u:(v,u) \in E} p_i(v, u) + \sum_{u:(u,v) \in E} p_i(u, v) > \theta\}$ for every topic i, and *node overlap coefficient* for topic i and j as $R_{ij}^V(\theta) = \frac{|\nu_i(\theta) \cap \nu_j(\theta)|}{\min\{|\nu_i(\theta)|, |\nu_j(\theta)|\}}$. If θ is small and the overlap coefficient is small, it means that the two topics are fairly separated in the network. In particular, we say that the network is *fully separable* for topics i and j if $R_{ij}^V(0) = 0$, and it is fully separable for all topics if $R_{ij}^V(0) = 0$ for any pair of i and j with $i \neq j$.

Algorithm 2 Marginal Influence Sort (MIS) Algorithm

```
Require: $G = (V, E)$, $k$, $\{p_i \mid i \in [d]\}$, $I = (\lambda_1, \cdots, \lambda_d)$, $\Lambda$, $S^g(k, \lambda p_i)$ and $MI^g(v, \lambda p_i)$, $\forall \lambda \in \Lambda$, $\forall i \in [d]$.
$I' = (\underline{\lambda}_1, \cdots, \underline{\lambda}_d)$
$V^g = \cup_{i \in [d], \underline{\lambda}_i > 0} S^g(k, \underline{\lambda}_i p_i)$
for $v \in V^g$ do
    $f(v) = \sum_{i \in [d], \underline{\lambda}_i > 0} MI^g(v, \underline{\lambda}_i p_i)$
end for
return top $k$ nodes with the largest $f(v), \forall v \in V^g$
```

Lemma 1. *If a network is fully separable among all topics, then for any* $v \in V$ *and topic* $i \in [d]$ *such that* $\sigma(v, p_i) > 1$, *for any item* $I = (\lambda_1, \lambda_2, \ldots, \lambda_d)$, *for any seed set* $S \subseteq V$, *we have* $MI(v|S, \lambda_i p_i) = MI(v|S, p)$, *where* $p = \sum_{j \in [d]} \lambda_j p_j$.

Lemma 1 suggests that we can use the marginal influence of a node on each topic when dealing with a topic mixture. Algorithm MIS is based on this idea.

Preprocess stage. Recall the detail of greedy algorithm, given probability p and budget k, for iteration $j = 1, 2, \cdots, k$, it calculates v_j to maximize marginal influence $MI(v_j|S_{j-1}, p)$ and let $S_j = S_{j-1} \cup \{v_j\}$ every time, and output $S^g(k, p) = S_k$ as seeds. Denote $MI^g(v_j, p) = MI(v_j|S_{j-1}, p)$, if $v_j \in S^g(k, p)$, and 0 otherwise. Therefore, $MI^g(v_j, p)$ is the marginal influence of v_j according to the greedy selection order. Suppose the landmark set $\Lambda = \{\lambda^c_0, \lambda^c_1, \lambda^c_2, \cdots, \lambda^c_m\}$. For every $\lambda \in \Lambda$ and every single topic $i \in [d]$, we pre-compute $S^g(k, \lambda p_i)$, and cache $MI^g(v, \lambda p_i)$, $\forall v \in S^g(k, \lambda p_i)$ in advance.

Online stage. Marginal Influence Sort (MIS) algorithm is described in Algorithm 2. Given an item $I = (\lambda_1, \cdots, \lambda_d)$, it first rounding down the mixture, and then use the union of seed sets as candidates. If a seed node appears multiple times in pre-computed topics, we approximate by summing the marginal influence in each topic together. Then we sort all candidates according to the computed marginal influence, and select top-k nodes as seeds.

Theorem 2. *Suppose* $I = (\lambda_1, \lambda_2, \cdots, \lambda_d)$, *where each* $\lambda_i \in \Lambda$, *and* $S^g(k, \lambda_1 p_1)$, $\cdots$, $S^g(k, \lambda_d p_d)$ *are disjoint. If the network is fully separable for all topics, the seed set calculated by Algorithm 2 is one of the possible sequences generated by greedy algorithm under the mixed influence probability* $p = \sum_{i \in [d]} \lambda_i p_i$.

Although MIS is a heuristic algorithm, this theorem implies that the seed set S from MIS satisfies $\sigma(S, p) \geq (1 - e^{-1} - \epsilon)\sigma(S^*, p)$ (for any $\epsilon > 0$) compared with the optimal S^* in fully separable networks. It suggests that MIS would work well for networks that are fairly separated among different topics, which are verified by our test results on the Arnetminer dataset. Moreover, even for networks that are not well separated, it is reasonable to assume that the marginal influence of nodes in the mixture can be approximated by the sum of the marginal influence in individual topics, and thus we expect MIS to work also competitively in this case, which is verified by our test results on the Flixster dataset.

4 Empirical Evaluation

We test the effectiveness of our algorithms by using multiple real-world datasets, and compare them with state-of-the-art influence maximization algorithms.

Data descriptions. The first dataset is on social movie rating network Flixster [2], an American social movie site for discovering new movies, learning about movies, and meeting others with similar tastes in movies. The Flixster network represents users as nodes, and two users u and v are connected by a directed edge (u, v) if they are friends both rating the same movie and v rates the movie shortly later after u does so. The network contains 29357 nodes, 425228 directed edges and 10 topics. We eliminate individual probabilities that are too weak ($\forall i \in [d], \lambda_i < 0.01$). We also obtain 11659 topic mixtures, from which we found that predominant ones are single topic (96.79%) or two-topic mixtures (3.04%). Mixtures with three or four topics are already rare and there are no items with five or more topics.

The second dataset is on the academic collaboration network Arnetminer [20], which is a free online service used to index and search academic social networks. The Arnetminer network represents authors as nodes and two authors have an edge if they coauthored a paper. It contains 5114 nodes, 34334 directed edges and 8 topics, and all 8 topics are related to computer science, such as data mining, machine learning, information retrieval, etc.

The above two datasets act as the baseline to verify the effectiveness of the algorithms. Furthermore, we use a larger academic collaboration network data DBLP maintained by Michael Ley (650K nodes and 2 million edges) only to test the scalability of the algorithms.

Influence probabilities. We first test our algorithms on the Flixster and Arnetminer datasets, whose influence probabilities are learned from real action trace data or node topic distribution data. The basic statistics for the learned influence probabilities show similar behavior between the two datasets, such as mean probabilities for each topic are mostly between 0.1 and 0.2, standard deviations (SD) are mostly between 0.1 and 0.3, etc. (Take the average over all topics: Arnetminer mean=0.173, SD=0.227; Flixster mean=0.131, SD=0.187.)

As DBLP does not have influence probabilities, we simulate two topics according to the joint distribution of topics 1 and 2 in the Flixster, and follow the practice of the TRIVALENCY model in [21] to rescale it into $\{0.1, 0.01, 0.001\}$ (i.e., strong, medium, and low influence).

Topic mixtures. In terms of topic mixtures, in practice and also supported by our data, an item is usually a mixture of a small number of topics thus our tests focus on testing topic mixtures from two topics. First, we test random samples to cover most common mixtures. We draw 50 topic mixtures from the uniform distribution over the polytope of any two topics. Second, since we have the data of real topic mixtures in Flixster dataset, we also test additional 50 cases following the same sampling technique described in Section 3.1 of [1], which estimates the Dirichlet distribution that maximizes the likelihood first and then generates topic mixtures by sampling from the distribution.

Algorithms for comparison. In our experiments, we test our topic-aware preprocessing based algorithms MIS and BTS comprehensively. Three classes of algorithms are selected for comparison: (a) Topic-aware algorithms: The topic-aware greedy algorithm (TA-Greedy) and a state-of-the-art fast heuristic algorithm PMIA (TA-PMIA) [21]; (b) Topic-oblivious algorithms: The topic-oblivious greedy algorithm (TO-Greedy), degree algorithm (TO-Degree) and random algorithm (Random); (c) Simple and fast heuristic algorithms that do not need preprocessing: The topic-aware PageRank (TA-PageRank) [4] and WeightedDegree (TA-WeightedDegree) [21] algorithms.

In this paper, we employ the greedy algorithm [15] with lazy evaluation and the same approximation ratio to provide hundreds of time of speedup to the original one [14]. PMIA is a fast heuristic algorithm based on trimming influence propagation to a tree structure, and it achieves thousand fold speedup comparing to optimized greedy algorithms with a small degradation on influence spread [21] (we set a small threshold $\theta = 1/1280$ to alleviate the degradation).

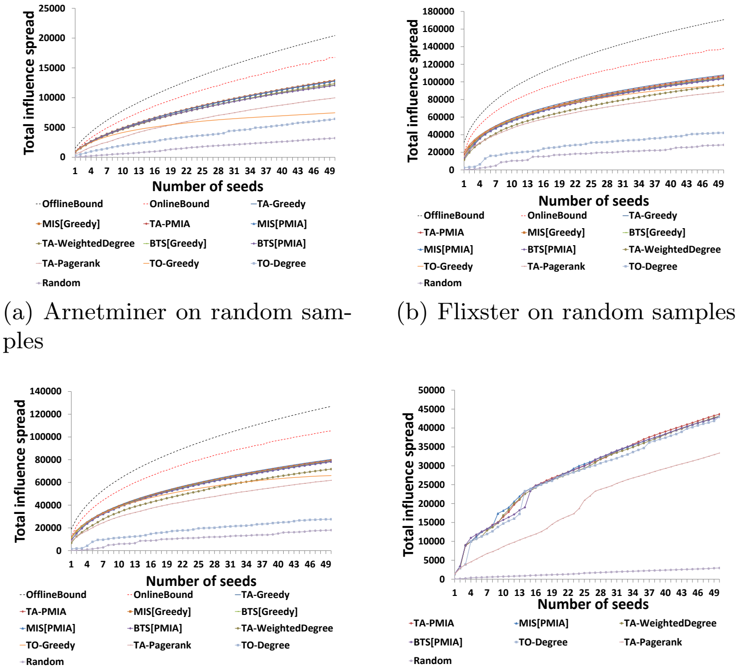
Topic-oblivious algorithms work under previous IC model that does not identify topics (the uniform topic mixture). TO-Greedy runs greedy algorithm for previous IC model. TO-Degree outputs the top-k nodes with the largest degree based on the original graph. Random simply chooses k nodes at random.

Finally, we study the possibility of acceleration for large graphs by comparing PMIA with greedy algorithm in preprocessing stage, and denote MIS and BTS algorithms as MIS[Greedy], BTS[Greedy] and MIS[PMIA], BTS[PMIA], respectively.

In the preprocessing stage, we use two algorithms, Greedy and PMIA, to precompute seed sets for MIS and BTS, except that for the DBLP dataset, which is too large to run the greedy algorithm, we only run PMIA. In our tests, we use 11 equally distant landmarks $\Lambda = \{0, 0.1, 0.2, \ldots, 1\}$ for MIS and BTS. Each landmark is independent and can be pre-computed concurrently in different processes. We choose $k = 50$ seeds in all our tests and compare the influence spread and running time, and take the average of 10000 Monte Carlo simulations to obtain the influence spread for each seed set in the greedy algorithm. In addition, we apply offline bound (the influence spread of any greedy seeds multiplied by factor $1/(1 - e^{-1})$) and online bound (Theorem 4 in [15]) to estimate influence spread of optimal solutions.

All experiments are conducted on a computer with 2.4GHz Intel(R) Xeon(R) E5530 CPU, 2 processors (16 cores), 48G memory, and Windows Server 2008 R2 (64 bits). The code is written in C++ and compiled by Visual Studio 2010.

Influence spread. Figure 1 shows the total influence spread results on Arnetminer with random samples (a); Flixster with random and Dirichlet samples, (b) and (c), respectively; and DBLP with random samples (d). For the Arnetminer dataset, it clearly separates all algorithms into three tiers (all percentages reported in parentheses are the gap of ratio compared with the best algorithm after taking average from one seed to 50 seeds): the top tier is TA-Greedy, TA-PMIA



(a) Arnetminer on random samples

(b) Flixster on random samples

(c) Flixster on Dirichlet samples

(d) DBLP on random samples

Fig. 1. Influence spread of algorithms. Legends are ordered (left to right, top to bottom) according to influence spread.

(0.61%), MIS[Greedy] (0.32%) and MIS[PMIA] (1.08%) whose gaps are negligible; the middle tier is TA-WeightedDegree (4.06%), BTS[Greedy] (4.68%), BTS[PMIA] (4.67%) and TA-PageRank (26.84%); and the lower tier is topic-oblivious algorithms TO-Greedy (28.57%), TO-Degree (56.75%) and Random (81.48%). Besides, MIS[Greedy] and BTS[Greedy] are 76.9% and 72.5% of the online bound, which are better than their conservative theoretical bounds ($1-e^{-1} \approx 63.2\%$). For Flixster dataset we see that the influence spread of TA-PMIA, MIS[Greedy], MIS[PMIA], BTS[Greedy] and BTS[PMIA] are 1.78%, 3.04%, 4.58%, 3.89% and 5.29% smaller than TA-Greedy for random samples, and 1.41%, 1.94%, 3.37%, 2.31% and 3.59% smaller for Dirichlet samples, respectively, indicating that our preprocessing based algorithms can perform quite well.

Running time. We summarize both of the preprocessing time and average online response time in Table 1. Table 1(b) shows the average online response time of different algorithms in finding 50 seeds (topic-oblivious algorithms always use the same seeds and thus are not reported). Our proposed MIS emerges as a strong candidate for fast real-time processing of topic-aware influence maximization task: it achieves microsecond response time, which does not depend on graph size or influence probability parameters, while its influence spread matches or is

Table 1. Running time statistics

(a) Preprocessing time

	Arnetminer ($\lvert\Lambda\rvert = 8 \times 11$)		Flixster ($\lvert\Lambda\rvert = 10 \times 11$)		DBLP ($\lvert\Lambda\rvert = 2 \times 11$)	
	Total	Max	Total	Max	Total	Max
Greedy	8.8 hrs	1.2 hrs	26.3 days	3.5 days	$\geq$ 100 days	$\geq$ 7 days
PMIA	37 secs	7.1 secs	2.28 hrs	12.6 mins	9.6 mins	4.2 mins

(b) Average online response time

	Arnetminer	Flixster		DBLP
		random	Dirichlet	
TA-Greedy	9.3 mins	1.5 days	20 hrs	N/A
TA-PMIA	0.52 sec	5.5 mins	3.8 mins	58 secs
MIS	2.85 µs	2.37 µs	3.84 µs	2.09 µs
BTS	1.20 µs	2.35 µs	1.42 µs	0.49 µs
TA-PageRank	0.15 sec	2.08 secs	2.30 secs	41 secs
TA-WeightedDegree	8.5 ms	29.9 ms	30.7 ms	0.32 sec

very close to the best greedy algorithm and outperforms other simple heuristics (Figure 1). Table 1(a) shows the preprocessing time based on greedy algorithm and PMIA algorithm on three datasets. It indicates that the greedy algorithm is suitable for small graphs but infeasible for large graphs like DBLP. PMIA is a viable choice for preprocessing, and our MIS using PMIA as the preprocessing algorithm achieves almost the same influence spread as MIS using the greedy algorithm for preprocessing (Figure 1).

5 Related Work

Domingos and Richardson [9,18] are the first to study influence maximization in an algorithmic framework. Kempe et al. [14] first formulate the discrete influence diffusion models including the independent cascade model and linear threshold model, and provide algorithmic results on influence maximization.

A large body of work follows the framework of [14]. One line of research improves on the efficiency and scalability of influence maximization algorithms [11,8,21,12]. Others extend the diffusion models and study other related optimization problems [5,3,13]. A number of studies propose machine learning methods to learn influence models and parameters [19,20,10]. A few studies look into the interplay of social influence and topic distributions [20,17,23,16]. They focus on inference of social influence from topic distributions or joint inference of influence diffusion and topic distributions. They do not provide a dynamic topic-aware influence diffusion model or study the influence maximization problem. Barbieri et al. [2] introduce the topic-aware influence diffusion models TIC and TLT as extensions to the IC and LT models. They provide maximum-likelihood based learning method to learn influence parameters in these topic-aware models. We use their proposed models and datasets with the learned parameters.

A recent independent work by Aslay et al. [1] is the closest one to our work. Their work focuses on index building in the query space while we use pre-computed marginal influence to help guiding seed selection, and thus the two approaches are complementary. Other differences have been listed in the introduction and will not be repeated here.

6 Future Work

One possible follow-up work is to combine the advantages of our approach and the approach in [1] to further improve the performance. Another direction is to study fast algorithms with stronger theoretical guarantee. An important work is to gather more real-world datasets and conduct a thorough investigation on the topic-wise influence properties of different networks, similar to our preliminary investigation on Arnetminer and Flixster datasets. This could bring more insights to the interplay between topic distributions and influence diffusion, which could guide future algorithm design.

References

1. C. Aslay, N. Barbieri, F. Bonchi, and R. Baeza-Yates. Online topic-aware influence maximization queries. In *EDBT*, 2014.
2. N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *ICDM*, 2012.
3. S. Bhagat, A. Goyal, and L. V. S. Lakshmanan. Maximizing product adoption in social networks. In *WSDM*, 2012.
4. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and Isdn Systems*, 30:107–117, 1998.
5. C. Budak, D. Agrawal, and A. E. Abbadi. Limiting the spread of misinformation in social networks. In *WWW*, 2011.
6. W. Chen, L. V. Lakshmanan, and C. Castillo. *Information and Influence Propagation in Social Networks*, volume 5. Morgan & Claypool, 2013.
7. W. Chen, T. Lin, and C. Yang. Real-time topic-aware influence maximization using preprocessing. *arXiv preprint arXiv:1403.0057*, 2014.
8. W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, 2009.
9. P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, 2001.
10. A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.
11. A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *WWW*, 2011.
12. A. Goyal, W. Lu, and L. V. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*, 2011.
13. X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, 2012.
14. D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.
15. J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.
16. C. X. Lin, Q. Mei, J. Han, Y. Jiang, and M. Danilevsky. The joint inference of topic diffusion and evolution in social communities. In *ICDM*, 2011.
17. L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, 2010.
18. M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.
19. K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES*, 2008.
20. J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, 2009.
21. C. Wang, W. Chen, and Y. Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *DMKD*, 25(3):545–576, 2012.
22. C. Wang, X. Yu, Y. Li, C. Zhai, and J. Han. Content coverage maximization on word networks for hierarchical topic summarization. In *CIKM*, 2013.
23. J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.