# Participation Maximization Based on Social Influence in Online Discussion Forums

# Microsoft Research Technical Report MSR-TR-2010-142 October 2010

Tao Sun[†],Wei Chen[‡],Zhenming Liu[§],Yajun Wang[‡],Xiaorui Sun[♯],Ming Zhang[†],Chin-Yew Lin[‡]
[†]Peking University. {suntao, mzhang}@net.pku.edu.cn
[‡]Microsoft Research Asia. {weic, yajunw, cyl}@microsoft.com
[§]Harvard School of Engineering and Applied Sciences. zliu@eecs.harvard.edu
[♯]Shanghai Jiaotong University. sunsirius@sjtu.edu.cn

## ABSTRACT

In online discussion forums, users are more motivated to take part in discussions when observing other users' participation — the effect of social influence among forum users. In this paper, we study how to utilize social influence for increasing user participation in online forums. To do so, we propose the use of sidebars, which display forum threads to users, as a mechanism to maximize user influence and boost participation. We formally define the participation maximization problem with the sidebar mechanism, based on the social influence network. We show that it is a special instance of the social welfare maximization problem with submodular utility functions and it is NP-hard. However, generic approximation algorithms for social welfare maximization is too slow to be feasible for real-world forums. Thus we design a heuristic algorithm, named Thread Allocation Based on Influence (TABI), to tackle the problem. Through extensive experiments using a dataset from a real-world online forum, we demonstrate that TABI consistently outperforms all other algorithms, including a personalized recommendation algorithm, in increasing forum participation.

The results of this work could facilitate other related studies such as designs for recommendation systems. The problem of participation maximization based on influence also opens a new direction in the study of social influence. Moreover, the proposed techniques can be applied to other social media, e.g., to maximize overall attention for advertisement in Facebook.

## Keywords

social networks, social influence, social welfare maximization, participation maximization, sidebar.

## 1. INTRODUCTION

The emergence of computer mediated communications has dramatically changed many people's social lives in the past decade. Among them, *online forums* have been serving as a major medium in the Internet that facilitates discussions of any kind. In an online forum, some discussions could be very specific (e.g., answering one particular question in Yahoo! Answers) while others could be more general (e.g., discussing travel experiences in TripAdvisor). Beyond the social value associated with the online forums, the owners of the forums also directly benefit from the traffic of healthy and active forums, e.g., more traffic means more advertising revenue.

At the individual level, when a user submits a new thread[1], besides accurate answers or valuable suggestions, s/he also hopes for a strong participation of other users in the thread. In fact, the users' psychological need of seeking attention exists in most social media, e.g., clip posters care about the number of views in YouTube, Twitter users care about their follower/retweet counts, and MySpace users care about the click through ratio in their "spaces". Thus being able to build a healthy and effective online forum platform that encourages users to participate in discussions would be beneficial to individual users as well.

To this date, albeit the much progress in system design that enables the building of large-scale and robust online forum platforms, only moderate progress has been made in the design of intelligent and automatic mechanisms that boost user participations into online discussions.

However, there have been several successful Q&A services that leverage information from social networking profiles to improve their platforms. For example, Aardvark[2] lets users

[1] A new thread means one user creates an initial post to start a new discussion in a forum.
[2] http://vark.com/

get real time answers from friends and friends-of-friends. Quora[3] concentrates on the quality of its users, instead of the quantity. It requires users to sign up with a Facebook or Twitter account, and collects authoritative responses from intelligent professional people. Facebook also rolled out its ambitious Q&A service “Questions” [16] in July 2010, which has been billed as Killer App considering its resource for the world’s largest social network — 500 million users. The success of the above services revealed one important idea: social ties among users have a positive effect on users’ posting behaviors.

Although in online discussion forums, there is typically no explicit social ties (i.e., friendship in Facebook), we observe that users tend to post after certain users — the effect of social influence, which can be viewed as implicit social ties. Inspired by this phenomenon, we propose strategies to increase participation based on influence among users. More specifically, we address this problem by delivering threads to forum users appropriately, so that discussion participation grows in a *measurable* way.

Delivering selected threads to users is similar in its form to current recommendation systems. In recommendation systems, usually the criteria of matching a thread with a user is whether the user’s friends also participate in the thread or whether there is any indication that the thread falls in the user’s interests, e.g. “Recommendation Engine” of Digg and “Recommended for You” of YouTube. While there are signs that the current practice of deciding recommendations is always beneficial to users, it is quite unclear how these isolated recommendations to individuals are impacting the ecosystem of forum as a whole. Nor is it clear whether making recommendations based sheerly on the users’ interests is optimal.

Hence, we design strategies for thread allocations to increase user participation based on social influence. Compared to personalized recommendation methods that focus on historical data to calculate which threads users will be most interested in, we further look into the future to maximize the forthcoming influence diffusion.

Existing models in maximizing influence diffusion by identifying a set of influential users [13, 4] do not fit well in the scenario of online discussion forums. For example, suppose that we identified the influential users and recklessly encouraged them to participate in every thread, the influential users would feel disturbed and find the recommendation unhelpful. As a result, for practicability, the notion of *budget* constraints (the number of threads to allocate to each user) is necessary in real forums. This leads to a new formulation of the optimization problem for online forums based on social influence — an optimal allocation problem to maximize overall participation through influence propagation.

More specifically, we first propose a stochastic user posting model for online forums, which is based on social influence propagation among the underlying social network. We then propose a sidebar scheme, in which each online user will be assigned a small number of threads in his/her sidebar, in order to increase the chance of his/her participation as well as the subsequent influence propagation to more users. The *particiation maximization* problem is the optimization problem of allocating threads to users’ sidebars to maximize the expected number of total participants in all threads.

We then prove that for any given thread, the expected number of total participants as a set function of users allocated with the thread is monotone and submodular. This characterizes the optimization problem as a specific instance of the *social welfare maximization* problem with submodular utility functions [5, 25], which suggests us to apply existing approximation algorithms in our setting.

However, these algorithms treats the utility function as an oracle, while in our case, evaluating expected number of participants given a sidebar allocation is very slow, rendering these approximation algorithms not feasible even for small-scale forums. Therefore, following the success of [4, 3] in dealing with a similar issue in influence maximization context, we turn to heuristic algorithms to achieve both efficiency and effectiveness for participation maximization. In this paper, we propose a heuristic algorithm, named *Thread Allocation Based on Influence(TABI)*, in which we explicitly consider both the factor of influence from the past in affecting the current user to post, and the factor of influence into the future for the current user to affect others.

We use data from a read-world online forum, TripAdvisor’s World travel forum[4], to evaluate our approach. We compare TABI with other algorithms including a personalized recommendation algorithm [23] and a social welfare maximization algorithm [5]. Our extensive simulation results clearly demonstrate that TABI performs consistently as the best algorithm in maximizing user participation.

Finally, we point out that our model and algorithms are not limited to online discussion forums. We believe that they have wider applicability to other social media context, and we discuss several new domains to apply our results.

To summarize, our contributions are mainly twofold:

*(i)* We propose the problem of participation maximization with the sidebar mechanism to utilize social influence for maximizing user participation in online forums, and connect the problem with the social welfare maximization problem; and

*(ii)* We propose an effective heuristic algorithm that beats existing recommendation algorithms and social welfare maximization algorithms empirically in maximizing participants in online forums.

## 2. RELATED WORK

In the context of online social media, there are many research works studying various aspects of social networks and social influence. We categorize the relevant works into several areas and briefly summarize them below.

**Effect of social ties on user behavior in social media.** Hogg and Szabo [12] used a stochastic approach to model users’ voting on Digg, and provided an explanation to the voting patterns. Hogg and Lerman [11] proposed an algorithm that described and predicted through iterative refinement how the popularity and interestingness of user-generated content evolved in time. In our context, we use a stochastic model to characterize user posting behaviors in online forums due to social influence among forum users.

**Learning social influence among individuals in the social network.** An important task in the study of social influence is to learn the strength of social influence among users from interactions. Gruhl et al. [9] used a variant of independent cascade model in blogspere and informally de-

[3] http://www.quora.com/

[4] http://www.tripadvisor.com/ForumHome

rived an Expectation-maximization(EM)-like algorithm to induce the influence probabilities among users. Saito et al. [20] derived a similar E-M algorithm in a more formal analysis to estimate influence probabilities. Goyal et al. [8] tackled the same problem in another variant of the influence propagation model, and applied Maximum Likelihood Estimator (MLE) instead of E-M algorithm to the Flickr social network. Influence analysis and learning supplies the soical influence graph as the input to the participation maximization problem, but itself is not the focus of our paper. We adapt the E-M algorithm of [20] to extract social influence in a real-world online forum TripAdvisor, and use it as input to our participation maximization algorithm.

**Application of social influence in social media.** This area is the most relevant one to our work. Extensive studies have been conducted to apply social influence in viral marketing [13, 14, 15, 17, 4, 3], personalized recommendation [23, 22, 6], ranking [22, 26], etc. However, as far as we know, there is no work studying how to increase participation through social influence in social media. Participation maximization differs from influence maximization studied in the context of viral marketing, because the latter focuses on finding a small set of influential users to maximize influence spread, while the former focuses on an appropriate allocation of the same amount of threads to every user to maximize overall participation. It also differs from personalized recommendation because the latter only focuses on suggesting the most relevant items to users to increase the chance of users accepting the items, but do not consider how users would influence other users to increase participation in the future. In Section 4, after formally defining the participation maximization problem, we will provide a more detailed comparison of participation maximization against influence maximization and personalized recommendation.

## 3. USER POSTING MODEL BASED ON SOCIAL INFLUENCE

In this section, we describe our model of user posting behavior in online discussion forums based on social influence among the users. Before providing the stochastic user posting model, we first describe the underlying social influence network.

A *social influence network* among the forum users is a directed and weighted graph $G = (\mathcal{U}, E, w)$, where $\mathcal{U}$ is the set of forum users, $E$ is the set of directed edges among these users, and $w$ is a weight function from the set of edges to real number in $[0, 1]$. The weight of an edge $(u, v) \in E$, referred to as the *influence probability* from $u$ to $v$ and denoted as $w_{u,v}$, indicates how likely user $u$ would influence user $v$ to write a post. As a convention, if $(u, v)$ is not an edge in $G$, we denote $w_{u,v} = 0$.

A forum $\mathcal{F}$ consists of its users $\mathcal{U}$, a set of threads $\mathcal{T}$, and sequences of posts generated by the users for every thread in the forum. We now describe the dynamic process of generating posts based on the social influence effect. To do so, we first augment the social influence graph $G$ by adding a *virtual user* $\tau$, together with edges from $\tau$ to all users in $\mathcal{U}$. We denote the extended influence network as $G_\tau = (\mathcal{U}_\tau, E_\tau, w)$, where $\mathcal{U} = \mathcal{U} \cup \{\tau\}$, $E_\tau = E \cup \{(\tau, u) \mid u \in \mathcal{U}\}$, and $w$ also contains weights for edges $(\tau, u)$ with $u \in \mathcal{U}$. Intuitively, the virtual user $\tau$ represents the content of the threads and its influence probabilities to users represent how the content of the threads affect users' posting behavior. Note that here $\mathcal{F}$ indicates one forum on a specific topic, U indicates users who participate in $\mathcal{F}$ and threads $\mathcal{T}$ of $\mathcal{F}$ is a group of threads with similar topics (e.g. all threads in the New York City category of the TripAdvisor's World travel forum). Thus we only introduce one virtual user for one $\mathcal{F}$, without adding different virtual users per thread.

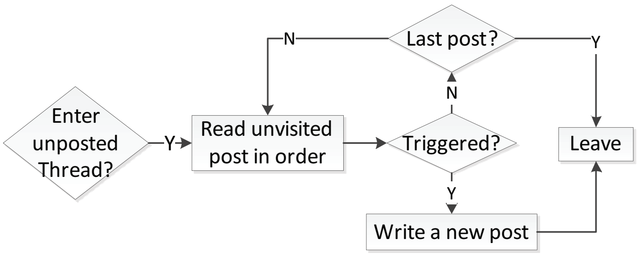

**Figure 1: Diagram on the user posting model for online discussion forums.**

We divide time into discrete time slots denoted as slot $0, 1, 2$, and so on. Slot 0 is a *virtual slot*, such that all threads in $\mathcal{T}$ are initialized with one post by the virtual user $\tau$ and no other users can post in this slot. At each slot $t \geq 1$, users in $\mathcal{U}$ may visit some threads, read the posts in the threads and perhaps write a post in some threads. In particular, each slot $t \geq 1$ is associated with a *visit probability* $\delta_t$, such that at slot $t$ each user has an independent probability of $\delta_t$ to visit each thread that s/he has not posted. For simplicity, we assume that all users are present in every slot. We could model users not online in all time slots by adding an online probability, but its treatment would be similar to the treatment of visit probability, and thus we ignored it in our paper.

Suppose that at slot $t \geq 1$ a user $v$ visits a thread $T \in \mathcal{T}$, and from the slot of $v$'s last visit in $T$ to slot $t - 1$, the sequence of existing participnats in $T$ is $u_1, u_2, \ldots, u_k$ (if $v$ never visited $T$ before, $u_1 = \tau$). Then at slot $t$, $v$ starts to read the posts by $u_1, u_2, \ldots, u_k$ in order. When $v$ reads the post by $u_i$, $v$ is influenced by $u_i$ to write a post in this thread $T$ with probability $w_{u_i,v}$. If $v$ has written a post in $T$, $v$'s revisits to $T$ are ignored, explained in more detail presently. A thread will eventually stop growing when (a) the visit probability becomes zero; or (b) all users have read all the existing posts in the thread but are not influenced to write one; or (c) all users have posted in the thread. Figure 1 shows the diagram of the user posting model.

We now provide some intuitive explanation and justification of our model.

**Social influence network.** The social influence network we defined is based on the Independent Cascade (IC) model for influence propagation defined in [13]. However, the dynamic model is different: IC model is for influence propagation in social networks starting from a seed set, while our model is for user appending posts to existing threads due to the social influence.

For our study of participation maximization, we consider the social influence network (with influence probabilities) as a given network. A number of researches provide methods in extracting the social network and influence probabilities [9, 1, 24, 19, 8]. In our experiment section (Section 6), we will adapt one of the methods to extract the social influence network from a real-world forum dataset, but this is not the focus of our paper.

**Visit probability.** Visit probability characterizes the likelihood of users entering threads based on the timeline of the threads. In the experiment section, we will see that in real-world forums, $\delta_t$ is often a fast decreasing sequence as $t$ increases, meaning that users are more likely to visit new threads and pay less attention to old threads, which is consistent with the observation in other social media [11].

**Single post vs. multiple posts.** In our model, we only record each user's first post in each thread, so that users's revisits to threads which they already participated are ignored as mentioned above. This simplification can be justified as follows. First, our participation maximization object is to maximize the number of distinct participants in the forum $\mathcal{F}$, not the number of posts generated, and thus multiple posts by a single user do not directly affect our optimization object. Second, if we want to model that multiple posts by a single user have an increased influence to other users, we could allow users to re-post, and model that each post of the user has the same and independent influence to other users reading the post. This is a direct extension of our model and our results still hold in this case. However, one may argue that repeated posts of a single user may not have the same and independent influence on other users, and this could make the model much more complicated. We left this extension as a future research item.

**Static threads vs. dynamic thread additions.** In our model, we do not explicitly incorporate dynamically adding threads in the forum $\mathcal{F}$, because threads added into $\mathcal{F}$ at different time slots are treated as different forum instances for the optimization purpose. More specifically, threads initiated at different time slots are separated from each other. Actually, they share the same optimization function as shown in Equation (1), so we can take threads $\mathcal{T}$ of one slot as a representative sample to elaborate our approach.

# 4. PARTICIPATION MAXIMIZATION WITH THE SIDEBAR MECHANISM

We propose a novel use of the sidebar mechanism based on social influence propagation to increase user participation into online discussion forums. We first introduce our sidebar mechanism and incorporate it into the user posting model to define the participation maximization problem. We then show that the expected number of participants of a thread with the sidebar mechanism has the submodularity property, making the participation maximization as an instance of social welfare maximization with submodular functions.

## 4.1 Problem formulation

A sidebar is a vertical bar on the side of a web page typically suggesting users with additional information. Sidebars are frequently used in many social media, such as the "People You May Know" sidebars in Linkedin and Facebook, and the "Top Answerers" sidebars in Yahoo! Answers.

In this paper, we propse to use sidebars in online discussion forums to suggest users with a small number of threads. The sidebar of a user increases the chance that the user visits his/her suggested threads. Then, by boosting the visit probability of these threads, we aim at increasing the chance that the user will post in these threads and in turn influence other users to post in them.

More formally, we define the *participation maximization* problem as follows. Each user has a budget constraint sidebar, which has space for display $B$ threads, where $B$ is a small constant (e.g. 5 or 10). At a certain time slot $s$, the system allocates $B$ threads from $\mathcal{T}$ to each user, so that the user would visit threads in his/her sidebar with a higher probability $\delta^*$. We use only one time slot for the allocations to sidebars for all threads in $\mathcal{T}$. Recall that in our model, all threads in $\mathcal{T}$ are modeled as initiated at the same virtual slot, and threads initiated at other time slots can be allocated at other time slots in the identical mechanism.

According to our user posting model, because visit probabilities to the threads shown in the sidebars are boosted to $\delta^*$, the mechanism can increase the probability that users posts in threads in their sidebars in succession, and in turn these posts may further influence subsequent users and increase the probability that others write posts in the thread. Thus, the overall number of participants in the forum $\mathcal{F}$(those who write posts) is increased.

Formally, let $S_j \subseteq \mathcal{U}$ be the set of users whose sidebars display thread $T_j$, and $\text{InfUser}^j(S_j)$ be the expected number of participants of $T_j$ after we display $T_j$ on the sidebars of a set of users $S_j$, calculated by our stochastic user posting model. Let $m = |\mathcal{T}|$ be the number of threads. Let $\mathcal{MU}$ be a multiset version of $\mathcal{U}$ such that each user $u \in \mathcal{U}$ appears $B$ times in $\mathcal{MU}$. Given as inputs (a) the social influence graph $G_\tau$, (b) a sequence of visit probabilities $\delta_j$'s, (c) thread set $\mathcal{T}$, (d) time slot $s \geq 1$ for sidebar allocation, (e) prefix of posts sequences up to slot $s-1$, (f) sidebar size $B$, (e) boosted visit probability $\delta^*$, the problem of participation maximization is to find a partition $\{S_1, S_2, \dots S_m\}$ of $\mathcal{MU}$ which maximizes

$$\sum_{j=1}^{m} \text{InfUser}^j(S_j),^5 \qquad (1)$$

which is the total (expected) number of participants in all threads.

The participation maximization problem defined above bear some resemblance to several related problems, but it also has its uniqueness. To further understand the problem, we compare it with several problems below.

**Comparison with recommendation systems.** Recommendation systems provide users with a small number of recommended items based on historical records of user actions and the assumption that users with similar activities in the past would be interested in similar items [23, 21]. In the context of online discussion forums, techniques in recommendation systems can certainly be used to assign threads to sidebars of interested users and potentially increase their participation. However, the key difference between recommendation systems and our participation maximization problem is that our problem is based on social influence among the users. More specifically, recommendation systems focus on predicting users' interests (e.g. who would be interested in which books or movies) to increase the chance that users accept the recommended items (e.g. the purchase of books or movie DVDs). In contrast, in our participation maximization problem, a good solution needs to recommend threads not only to the users who are likely to post in these threads, but also to the users who are likely to

[5] Since $\text{InfUser}(\cdot)$ is not defined on multisets, we simply ignore additional copies of the same user in $S_j$. Also some $S_j$'s could be empty, meaning that these threads are not assigned to any sidebars.

influence others to post. This is because our optimization object is to maximize the total participation, not just the number of posts immediately caused by sidebar recommendations. Considering the future influence generated by the sidebar recommendations is the novelty differentiating our work from other recommendation systems.

In Section 6, we empirically compare a solution we proposed with a personalized recommendation method [23], which models information diffusion, and thus bearing some resemblance to our influence propagation model. Our results show that our solution outperforms the recommendation system because we look into the future influence propagation.

**Comparison with influence maximization for viral marketing.** Influence maximization in the context of viral marketing have been extensively studied recently [13, 14, 15, 17, 4, 3]. The problem is to find a small seed set in a social network to maximize their eventual influence spread. In the context of online discussion forum, if our goal is only to maximize the number of participants for a specific thread, and we have a constraint on the number of users to be selected for promoting the thread, then it falls into the domain of influence maximization problem. However, we aim at maximizing the *overall* number of participants among all the threads, and the constraint is *not* on the number of users each thread can be recommended to, but on the number of threads each user can be recommended. Hence, the problem formulation becomes markedly unlike influence maximization, and thus requires different solutions.

**Comparison with social welfare maximization.** The participation maximization problem can be viewed as an instance of the social welfare optimization problem [5, 25], in which resources are allocated to consumers, who have certain utility for every combination of the resources, and the goal is to maximize the total utility of all consumers. In the context of online discussion forums with sidebars, panels in sidebars can be viewed as resources and threads as consumers, and the utility function of thread $T_j$ is exactly $\mathrm{InfUser}^j(S_j)$. With the submodularity property on $\mathrm{InfUser}^j(\cdot)$ as proved in Section 4.2, participation maximization is a specific instance of social welfare maximization with submodular utility functions, for which a number of theoretical studies have provided approximation algorithms [5, 25]. However, these algorithms treat utility functions as an instant oracle, while for real forums, the calculation of $\mathrm{InfUser}^j(\cdot)$ is too complicated and time-consuming to be feasible in practice, as to be shown in our experiment section (Section 6). Therefore, we need to design specific algorithms for our participation maximization problem.

## 4.2 Submodularity of $\mathrm{InfUser}^j(\cdot)$

Function $\mathrm{InfUser}^j(\cdot)$ satisfies an important property called *submodularity.* A set function $f$ on $\mathcal{U}$ is submodular if for any set $S, T \subseteq \mathcal{U}$, we have

$$f(S) + f(T) \geq f(S \cap T) + f(S \cup T).$$

Moreover, a set function $f$ on $\mathcal{U}$ is monotone if for all $S \subseteq T \subseteq \mathcal{U}$, $f(S) \leq f(T)$. For set function $\mathrm{InfUser}^j(\cdot)$, we have

Theorem 1. *The function* $\mathrm{InfUser}^j(\cdot)$ *is monotone and submodular, for all* $j \in [m]$.

Proof. (Outline). It is similar to the proof of submodularity of the original influence function in [13]. However, we have to address the challenge in our model: encoding more random events, in particular, the visiting events and the influence propagation events. Therefore, we build a graph consisting of multiple levels. Each level represents the influence social network at a particular time slot. The visiting event of each node are encoded by a random coloring process. Then the influence function is simply counting the number of reachable nodes with a particular color from a seed set, which is clearly submodular.

The detailed proof is included in Appendix A. □

# 5. THREAD ALLOCATION ALGORITHMS

In this section, we discuss thread allocation algorithms for our sidebar mechanism, and propose our heuristic algorithm TABI as an effective and efficient solution to the participation maximization problem.

Due to the combinatorial nature of the problem, one cannot enumerate all possible allocations to find the optimal solution. In fact, we show that it is NP-hard to find the optimal allocation.

Theorem 2. *Finding the optimal solution to the participation maximization problem is NP-hard, even if there are only two threads in the forum and computing* $\mathrm{InfUser}^j(S)$ *for any* $S \subseteq \mathcal{U}$ *is a polynomial-time task.*

Proof. (Outline). The proof is by a reduction from the MaxCut problem. The complete proof is included in Appendix B. □

Now we discuss several approaches to overcome the NP-hardness result.

**Random allocation.** The most straightforward approach is to allocate threads to sidebars uniformly at random. In general, random allocations would not perform well, but in a special case to allocate threads as soon as they are generated ($s = 1$ in our model, since $s = 0$ is a virtual slot), it is indeed an approximation algorithm. More specifically, when $s = 1$, all threads in $\mathcal{T}$ only have the same initial post by the virtual user $\tau$, and thus the utility functions $\mathrm{InfUser}^j(\cdot)$ are the same for all threads, in which case Vondrák [25] proved that random allocation is a $(1 - 1/e)$-approximation algorithm. Moreover, Vondrák pointed out that this approximation is tight when utility function evaluation is given as an oracle. Even though in our case the utility function $\mathrm{InfUser}^j(\cdot)$ is not an oracle, it still indicates that it is not likely to beat the simple random allocation for the special case of $s = 1$.

However, when $s \geq 2$, most threads already have some posts (written by users at slot 1) and they are likely to be different. This causes the utility function $\mathrm{InfUser}^j(\cdot)$ to be different among the threads, and random allocation is no longer a good choice. Our simulation results will show that it is indeed the case.

**Approximation algorithms, in particular Randomized Proportional Allocation (RPA) algorithm of [5].** As proved in Theorem 1, the utility function $\mathrm{InfUser}^j(\cdot)$ is monotone and submodular, thus approximation algorithms for the general social welfare maximization problem with submodular functions [5, 25] can be applied to solve the participation maximization problem. Algorithm 1 presents our adaptation of a $(2 - \frac{1}{m})$-approximation algorithm [5], where $m$ is the number of threads in our model. Essentially, the algorithm computes the incremental effect $R_j$ of assigning

**Algorithm 1** Approximation Algorithm

```
1: /* n users, m threads, $P_v$ is the constraint panel number for each $v$*/
2: initialize $P_v = B$ for all $v \in \mathcal{U}$, $S_j = \emptyset$ for all $j \in \mathcal{T}$
3: for each $v \in \mathcal{U}$ with $P_v > 0$ do
4:     for each $j \in \mathcal{T}$ do
5:         $R_j = \text{InfUser}^j(\{v\} \cup S_j) - \text{InfUser}^j(S_j)$
6:     end for
7:     select exactly one thread $j$ randomly as follows: each thread $j$ is chosen with probability $\frac{R_j^{m-1}}{\sum_{T_k \in \mathcal{T}} R_k^{m-1}}$
8:     update $S_j = S_j \cup \{v\}$ and $P_v = P_v - 1$.
9: end for
```

thread $T_j$ to user $v$, given that $T_j$ has already been assigned to a set of users $S_j$ (line 5), and then pick a thread $T_j$ at random with a probability proportional to $R_j^{m-1}$ (line 7). We select this algorithm because of its simplicity and it supports online computation — the computation of assigning threads to a user's sidebar could be done for the user when s/he is online, independent of assignments of users who logs in later.

However, RPA as well as other approximation algorithms has a serious drawback. It assumes that te computation of utility function is done by an oracle, but in real forums, it is difficult to compute $\text{InfUser}^j(S)$. Similar to the case of calculating influence spread in influence maximization [13, 4], sufficient amount of simulations are required to obtain a relatively accurate estimate of $\text{InfUser}^j(S)$. In our case the algorithm would be even slower because we have $m$ (=number of threads) different submodular utility functions for each user to evaluate, and it is difficult to apply optimization techniques [4] under the setting. Our experimental results in the next section show that the RPA algorithm is very slow and preforms poor under insufficient number of simulations. This leads us to consider fast heuristic algorithms to tackle the problem.

**Our heuristic algorithm: Thread Allocation Based on Influence (TABI).** We propose TABI, a heuristic algorithm to solve the participation maximization problem. The idea of TABI is to estimate the incremental effect of allocating thread $T_j$ to a user $v$ by a fast neighborhood calculation.

Let $EP_j$ denote the set of Existing Participants in thread $T_j$ before the allocation time slot $s$. Let $I_v$ and $O_v$ denote the set of $v$'s in-neighbors and out-neighbors in the influence graph $G_\tau$, respectively. The probability that $v$ is influenced by at least one of its in-neighbors in $EP_j$ is $(1 - \prod_{u \in EP_j \cap I_v} (1 - w_{u,v}))$. Provided that $v$ is influenced, the expected number of additional users would include

*(i)* $v$ itself, with probability 1.

*(ii)* each of $v$'s inactive out-neighbor $x, x \in O_v \setminus EP_j$, who would be influenced by $v$ rather than any users in $EP_j$, with probability $w_{v,x}(\prod_{u \in EP_j \cap I_x} (1 - w_{u,x}))$.

Thus, the additional users $\Delta\text{Inf}_v^j$ that brought by displaying thread $T_j$ to $v$ is estimated as:

$$(1 - \prod_{u \in EP_j \cap I_v} (1 - w_{u,v}))(1 + \sum_{x \in O_v \setminus EP_j} w_{v,x} \prod_{u \in EP_j \cap I_x} (1 - w_{u,x})) \tag{2}$$

Once the estimates are obtained on all threads, we rank these estimates and select the top $B$ threads to allocate to user $v$ (Algorithm 2). Notice that $\delta^*$ is the same value for all $v \in \mathcal{U}$ if we display $T_j$ to $v$, so we don't have to multiply the $\Delta\text{Inf}_v^j$ by $\delta^*$ for ranking and selection.

**Algorithm 2** TABI

```
for each $v \in \mathcal{U}$ do
    for each $j \in \mathcal{T}$ do
        calculate $\Delta\text{Inf}_v^j$ as Equation 2
    end for
    Rank threads by $\Delta\text{Inf}_v^j$ in descending order
    Select top B threads to display in $v$'s sidebar
end for
```

The above estimate contains two parts: *(i)* the first parenthesis, which captures how likely the user $v$ is influenced by existing participants; and *(ii)* the second parenthesis, which captures how likely $v$ will influence other users in the future. Conceptually, the first part is similar to a recommendation system, while the second part focuses on incorporating future influence into thread selection, which we believe is our unique consideration differing from recommendation systems. The estimation in TABI is simplified, without considering further influence cascades and visit probabilities in the future slots. Nevertheless, the simulation results will show that the performance of TABI already beats other algorithms.

# 6. EXPERIMENTS

In this section, we use data from a real-world online discussion forum to evaluate the effectiveness of our TABI algorithm and compare it against several other algorithms. We first extract the parameters, such as the social influence graph and visit probabilities from the forum data, and then simulate different algorithms in our user posting model with these parameters to compare the expected number of participants they achieve.

## 6.1 Datasets

Our datasets are crawled from TripAdvisor's World travel forum. With more than 20 million monthly visitors and 6 million registered members, TripAdvisor represents the largest travel community in the world. TripAdvisor forum is discussion oriented, where users share candid opinions, hotel reviews, traveling experience or raise questions and discuss possible solutions. It consists of a number of discussion categories (one $\mathcal{F}$ for one category in our model) typically separated by locations. To conduct the experiments, we select three most active categories in TripAdvisor, which are Orlando, London and New York City (NYC).

Even though we have crawled data for several years, most users will only have a short active period on social media [10, 18]. The influence among users are also likely to change over time. Thus we use data from a relatively short period to guarantee active users and their stable influence relationship. However, if the period is too short, many social interactions and influence relationship will be missing. Hence, an appropriate time period should be carefully selected.

In our experiment, we calculated a user's *forum life span* as the time period between her first and last post on a forum. We then choose a window size *t_win* such that around 80% of users have their forum life spans within *t_win*. In TripAdvisor, *t_win* is about 60 days. Thus, we choose a 60 day period in the beginning of year 2009 for our experiments.

**Table 1: Statistics of the dataset from TripAdvisor**

| Category | Orlando | London | NYC |
|---|---|---|---|
| Time Period | 2009.1.1 - 2009.3.1 (60 days) | | |
| threads number | 4,062 | 1,800 | 2,455 |
| users number | 2,085 | 1,467 | 1,694 |
| Avg postsNum/thread | 5.16 | 5.34 | 4.91 |
| Max postsNum/thread | 63 | 67 | 79 |
| Avg postsNum/user | 10.06 | 6.56 | 7.11 |
| Max postsNum/user | 1,142 | 539 | 1,239 |

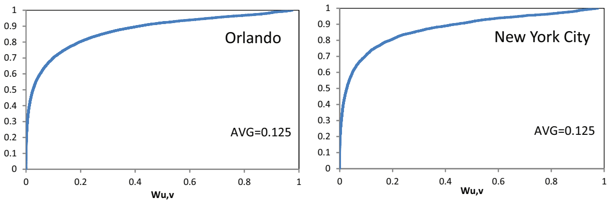

**Figure 2: CDF of influence probabilities**

The basic statistics of this dataset are given in Table 1. Recall that if a user posts more than once on a same thread, we only record her first post.

## 6.2 Extracting the social influence network

In the formulation of the participation maximization problem given in Section 3, the social influence network is treated as an input of the problem. In the case of TripAdvisor, no explicit social network is maintained, and we need to extract an implicit influence network as well as learning the influence probabilities on the network.[6]

For constructing the network, intuitively, if the posts of one user influence another user and lead to his/her posting on the same thread, there will be a link from the first user to the second user. Thus in the influence graph $G_\tau = (\mathcal{U}_\tau, E_\tau, w)$, we keep edge $(u, v)$ iff $v$ follows $u$ to post in at least $N$ threads ($N = 2$ in our experiment).

There are several studies on learning the influence probabilities in a network [9, 20, 8, 19]. Based on our forum context, we adapt the E-M algorithms in [9, 20] to fit into our user posting model as described in Section 3. Roughly speaking, to calculate $w_{u,v}$'s, the algorithm iterates between two conditional probabilities: *i)* in threads that $v$ posts after $u$, compute the conditional probability that $v$ posts because of $u$'s influence given $v$ posts in $T_j$. *ii)* update $w_{u,v}$ by estimating the probability that $v$ is influenced by $u$ given $v$ reads $u$'s post. The algorithm converges after a number of iterations, at which we obtain $w_{u,v}$ on each directed edge $(u, v)$. To avoid cluttering the main flow of our paper, our detailed social influence graph learning algorithm is given Appendix C. The Cumulative Distribution Function (CDF) distributions are given as Figure 2 (CDF of category London with similar distribution is omitted here).

## 6.3 Estimating visit probabilities

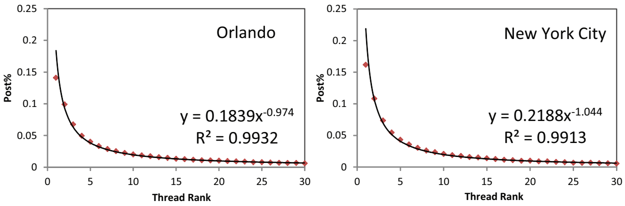

**Figure 3: Post distribution v.s. thread rank**

Since our TABI algorithm does not depends on visit probabilities, we would like to test the algorithm against different visit probability sequences. Meanwhile, we also want to obtain a visit probability sequence that is similar at least in trend to the real data. However, since we cannot get access to the login and browsing data of TripAdvisor users, accurate estimation of visit probabilities is not feasible. Therefore, we make use of users' posting data that we crawled to approximate visit probabilities.

Following Hogg and Lerman's definition of *recency* [12], we estimate visit probability from posting data based on *thread rank r*. Thread rank r of $T_j \in \mathcal{T}$ at a time $t$ is defined as: its rank in chronological order of all threads at $t$. For example, at $t$, among all $m$ threads, the latest (submitted most recently) thread has $r = 1$, while the oldest has $r = m$. We want to estimate the visit probability $\delta_r$ for threads with rank value $r$. Every time there is a post in $T_j$, the post can be assigned with $T_j$'s thread rank value $r$. Then $\delta_r$ is proportional to the ratio between the number of posts with $r$ and the total number of posts among all threads.

Figure 3, marked with power-law trend line, shows how the proportion of posts varies with thread rank in Orlando and NYC of TripAdvisor. Both curves fit very well into power-law curves, with power-law exponents $\alpha$ being $-0.974$ and $-1.044$ respectively. Results in other categories show similar power-law distributions. We anticipate that the visit probabilities would have a similar power-law trend, which coincides with our intuition that people pays a lot more attention to recent threads than earlier threads but there is always some people visiting old threads.

In our simulation, we use $\delta_r$ to approximate the visit probability $\delta_t$, and we further try different visit probability sequences, one from a different power law curve and the other a constant sequence.

## 6.4 Simulation tests and results

Since we cannot deploy our proposed sidebar mechanism in a real online forum environment, we demonstrate the effectiveness of the mechanism via simulations based on the user posting model and the parameters we have analyzed. In our simulation, for simplicity, we assume that every user is online for a period of time in every time slot so that they have a chance to visit each thread. In each category, there are $n$ users ($n = |V|$ in $G_\tau$), $m$ threads ($m = |\mathcal{T}|$) and sidebar budget $B$ ($B = 5$ in all the following experiments).

We simulate five participation maximization algorithms:

*i)* NoSidebar, as the baseline;

*ii)* Random, allocation with the uniform probability distribution;

*iii)* Randomized Proportional Allocation (RPA) [5]. The adapted version as described in Algorithm 1;

*iv)* TEABIF, a personalized recommendation algorithm,

[6] TripAdvisor recently introduced "Trip Friends" feature [2] in partnership with Facebook, but Facebook network does not necessarily coincide with the influence network among TripAdvisor users. Our approach of extracting social influence network from social interaction is also more general.

called topic-sensitive early adoption based information flow (TEABIF) [23], which recommends items to users by estimating whom the information will propagate to with high probabilities. We select TEABIF as a representation of recommendation systems because it considers information diffusion in their model, which has some similarity with our influence propagation model. TEABIF (as well as other recommendation systems), however, does not exactly fit into our participation maximization formulation, since it does not take a social influence graph as the input. Instead, we directly feed TEABIF with the post sequence data used in our test to derive its recommendations.

*v)* TABI, as described in Algorithm 2.

**Comparing the effectiveness of different algorithms.** In our first test, we compare the effectiveness among the above five algorithms with the following simulation setup. We run tests for different thread number $m = 30$, 40, and 50 respectively. We use $\delta_r$ described in Section 6.3 directly as the visit probabilities, and set the boosted visit probability $\delta^* = 0.8$. Notice that the value of $\delta^*$ would not affect thread allocation of Random, TEABIF and TABI, and thus total participation has a linear relationship with $\delta^*$. For RPA, its thread allocation depends on $\delta^*$ when calculating $\text{InfUser}^j(S)$, but our simulation results show that total participation is still close to a linear relationship with $\delta^*$. Therefore, results for other $\delta^*$ values only have a constant factor difference and can be derived, so we do not report the exact numbers here.

In the simulations, we pre-populated existing participants $EP_j$ of each $T_j$ in slot 1 based on the user posting model as Figure 1, and each such pre-population is called a *group*, a.k.a, $\bigcup_{T_j \in \mathcal{T}} EP_j$. In slot 2, we use different algorithms to perform thread allocation (i.e., $s = 2$ for this test). For each simulation, we run it from slot 2 to slot 15 based on the posting model, and collect the additional participants (or newParticipants, who post after slot 1). For each group, we run 1000 simulations to obtain the average number of newParticipants. For each category, we generate 500 independent groups and take the average as the final number reported in our result.

The RPA algorithm would be extremely slow if we also run 1000 simulations to obtain one $\text{InfUser}^j(S)$ value in Algorithm 1. To finish RPA in a reasonable amount of time, we run 10 simulations to estimate $\text{InfUser}^j(S)$. Even in this case, RPA still takes hours to finish one group, while all other algorithms only take seconds. Thus for RPA, we have to compromise and collect average value from 50 groups, instead of 500 groups. It demonstrates that the RPA (and other social welfare maximization algorithms based on utility oracles) cannot be used in practice, where we need efficient and online computations for thread allocations.

The results of this test for category NYC, London and Orlando are given in Figure 4. In all nine tests covering three categories and three different numbers of threads $m$, our TABI algorithm performs consistently as the best algorithm. Comparing to TEABIF, take $m = 40$ as the example, the improvement of TABI over TEABIF in NYC, London and Orlando are $19.87 \pm 6.32$,$20.13 \pm 6.51$,$27.52 \pm 9.01$, respectively, corresponding to percentage increases of 6.2%, 5.7%, 5.5% respectively, and all improvements are statistically significant. RPA algorithm performs worse than TABI and TEABIF, which can be partly attributed to insufficient number of iterations trading accuracy for efficiency. Comparing to NoSidebar and Random, TABI significantly outperforms both of them, with a large margin of 50-60% and 30-40%, respectively. It indicates that sidebar mechanism with our TABI algorithm could significantly increases participation, comparing with the case of no sidebars or randomly targeted sidebars.

**Effectiveness on different visit probabilities.** In this test, we intend to see if our TABI algorithm could perform consistently better than other algorithms under different visit probability sequences. To do so, we repeat the above test with the following two visit probability sequences:

*i)* Power law: $\delta_t = kt^{-\alpha}$, with $k = 0.3$ and $\alpha = 0.6$, to simulate the decreasing trend with a larger visit probability values compared to $\delta_r$.

*ii)* Constant value: $\delta_t = 0.1$ for all $t$.

Figure 5 shows the result with threads number $m = 40$ in all the three categories. In the first test, RPA approximation algorithm has already been shown to be exceedingly time consuming and ineffective , so RPA is excluded here. We can see that under both visit probability sequences, TABI's improvement over TEABIF and other methods are consistent.

**Effectiveness on different allocation time slots.** In this test, we aim at checking whether TABI could perform consistently better than other algorithms under different allocation time slot $s$. To this end, we vary $s$ from 2 to 10, set $m = 40$ and $\delta^* = 0.5$. $\delta_r$'s is used as the visit probabilities.

We would also like to compare the total participation (from slot 1 to slot 15) between different allocation time slots, not the additional number of participants (newParticipant) after the sidebar allocation slot $s$. Since different allocation time slots have different set of pre-existing participants, we set up the test in the following steps to put the results of different allocation slots under the same scale.

*i)* Pre-populated one group of existing participants in the same way as described in the first test.

*ii)* For each group, if sidebar allocation is at slot 2 ($s = 2$), we simply run simulations 1000 times until slot 15, and take the average of the number of participants. If sidebar allocation is at slot $s > 2$, we first prepopulate users in slots 2 to $s-1$ based on user posting model, then run simulations 1000 times with allocation at slot $s$. We use 500 prepopulation sets (slots 2 to $s-1$) to take the average of results for $s > 2$, in order to make a fair comparison with the case of $s = 2$.

The above two steps is for one group of existing participants at slot 1. And we run 100 independent groups to take the average as the final result, which is reported in Figure 6.

Our results show that TABI always outperforms TEABIF and Random in all the allocation time slot $s$, which means that TABI works well with different existing participants and different future visit probability sequences. We also notice the increasing trend of participation as $s$ increases. From this, one may be tempted to conclude that we need to use sidebars for "older" threads. However, we need to take such conclusion cautiously. The reason of the increasing trend is mainly due to the fact that the visit probability sequence is a decreasing sequence, and thus in later slots threads receive a larger boost in visit probabilities when shown in the sidebars. However, recommending "older" threads may result in bad user experiences. Therefore, we believe a

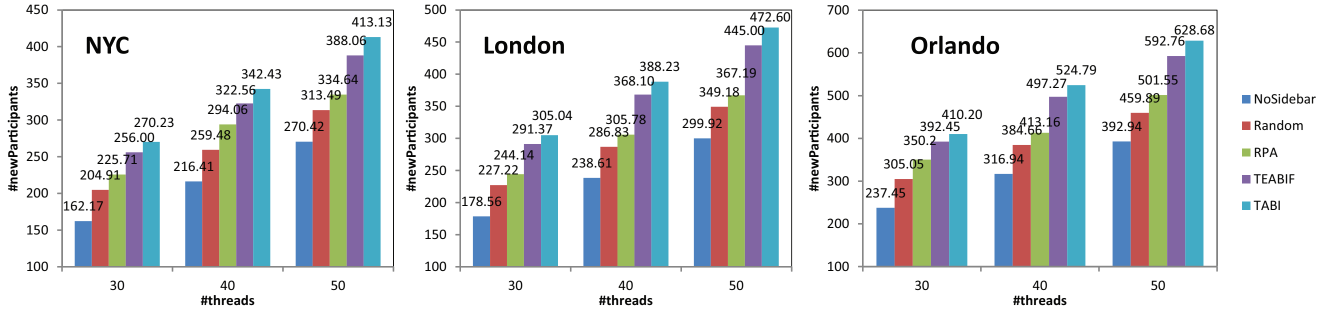

Figure 4: Results of Five Approaches

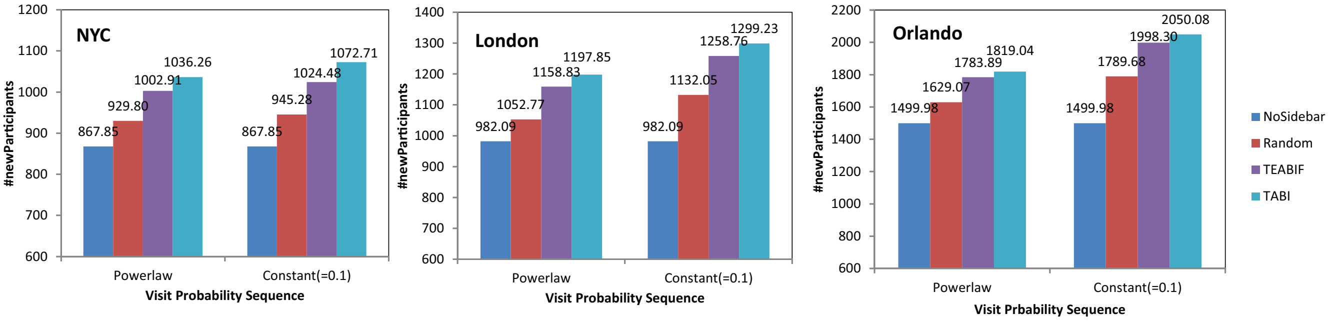

Figure 5: Results on different visit probability sequences

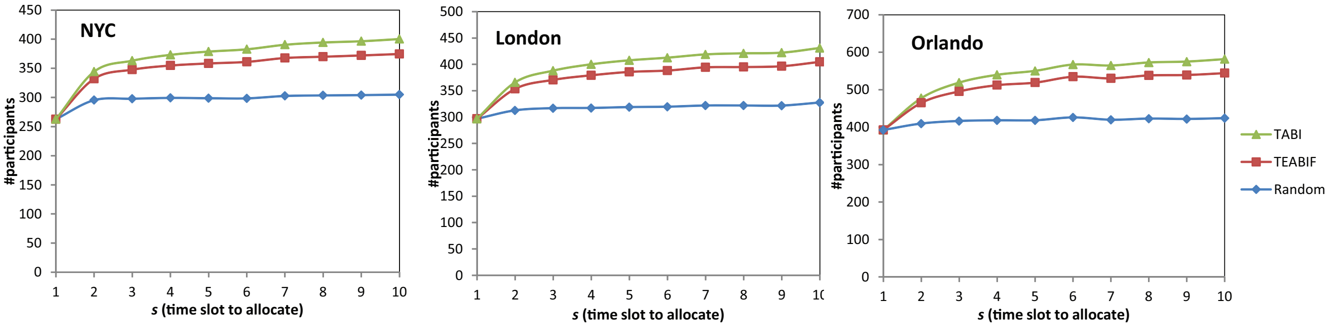

Figure 6: Results on different allocation time slots

better conclusion is that larger boost in visit probabilities may provide more participation, but the selection of time slot $s$ to use sidebars should consider other factors such as overall user experiences. This is why we use $s$ as a parameter of the problem rather than a variable to be tuned in the optimization process.

To summarize, our simulation results clearly demonstrate that sidebar mechanism based on social influence can significantly improve participation, and in all situations our algorithm TABI performs the best over all other algorithms, including an approximation algorithm and a personalized recommendation algorithm.

# 7. DISCUSSION AND CONCLUSION

## 7.1 Discussion

The proposed sidebar mechanism with TABI algorithm can also be applied to other social media, with the purpose to maximize overall participation, activity or attention. The target medium should satisfy the following characteristics: *i)* users own unique user IDs; *ii)* users are able to observe other users' behaviors on one specific event; *iii)* users can interact with each other. Here, we give several examples as below.

**Advertisement in Facebook.** Display B promotions in each user's sidebar to maximize the overall number of audience in Facebook. Each advertiser has its own Facebook page and writes updates to gain attention, e.g., companies as Starbucks to distribute coupons, or TV shows as American Idol to announce official news for high ratings. If the targeted users visit the page and click on the "Like" button or comment on the advertiser's updates, a link to the advertiser gets added to their Facebook stream. Then, their friends have a chance to notice the link, if they also click "Like" or comment, friends of friends can see the link. Note that the users are able to browse "People who like this" or view all comments in Facebook, which is similar to browsing people who already posted to one thread in online discussion forums. In this way, the promotion spreads and reaches a wide audience. Our mechanism can allocate the right promotions to the right users to maximize the overall audience.

**Posts in Google Buzz.** Display B buzzes in each user's sidebar to maximize the overall attention. In Google Buzz, users can also follow others, and reshare, like, or comment on followees' shared posts. And they can also view who reshare, like, or comment on the which specific posts. Thus it is similar to the case of online discussion forums. And the overall attention of Buzz gets maximized trough our sidebar mechanism.

**Video Comments in YouTube.** Display B video links in each user's sidebar to maximize the overall comments. In YouTube, users can comment on one video, and others who are also interested in the clip can read all the previous comment and discuss with them. Thus, comments on video is similar to posts on one threads, and the uploaders always expect lively discussions. By calculating influence among users and adding the sidebar mechanism, our approach can help to bring up the level of discussion on the site.

## 7.2 Conclusion

To summarize, in this paper, we propose the sidebar mechanism to maximize participation based on social influence in online discussion forums. We formulate the problem as participation maximization problem, a special case of social welfare maximization problem. We prove that it is NP-hard, and it has the property of monotonicity and submodularity. In real applications, in order to overcome the inefficiency of previous approximation algorithms, we propose a heuristic algorithm TABI for thread allocation. Through extensive simulations, we validate the robustness and effectiveness of TABI. The whole approach can also be applied to other social media to increase total participation.

For future work, we will investigate heuristics that consider further influence cascades and find out the best timing for thread allocation. Besides, we will study the application of similar approaches to other social media, which may have rich interaction and social network data.

# 8. REFERENCES

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *Proc. of KDD*, pages 7–15, 2008.

[2] L. Bly. Tripadvisor lets users tap facebook friends for advice. *USA TODAY*, Jun 14, 2010.

[3] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large scale social networks. In *Proc. of KDD*, 2010.

[4] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proc. of KDD*, pages 199–208, 2009.

[5] S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proc. of SODA*, pages 1064–1073, 2006.

[6] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *Proc. of KDD*, pages 289–298, New York, NY, USA, 2009. ACM.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[8] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proc. of WSDM*, pages 241–250, 2010.

[9] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proc. of WWW*, pages 491–501, 2004.

[10] L. Guo, E. Tan, S. Chen, X. Zhang, and Y. E. Zhao. Analyzing patterns of user content generation in online social networks. In *Proc. of KDD*, pages 369–378, 2009.

[11] T. Hogg and K. Lerman. Stochastic models of user-contributory web sites. In *Proc. of ICWSM*, 2009.

[12] T. Hogg and G. Szabo. Diversity of user activity and content quality in online communities. In *Proc. of ICWSM*, pages 58–65, 2009.

[13] D. Kempe, J. Kleinberg, and Èva Tardos. Maximizing the spread of influence through a social network. In *Proc. of KDD*, pages 137–146, 2003.

[14] D. Kempe, J. Kleinberg, and Èva Tardos. Influential nodes in a diffusion model for social networks. In *Proc. of ICALP*, pages 1127–1138, 2005.

[15] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *Proc. of ECML PKDD*, pages 259–271, 2006.

[16] J. Kincaid. Facebook Q&A service 'Questions' begins rolling out, could be massive, Jul 28, 2010. http://techcrunch.com/2010/07/28/facebook-qa-service-questions-begins-rolling-out-could-be-massive/.

[17] H. Ma, H. Yang, M. R. Lyu, and I. King. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proc. of CIKM*, pages 233–242, 2008.

[18] K. K. Nam, M. S. Ackerman, and L. A. Adamic. Questions in, knowledge iN?: a study of naver's question answering community. In *Proc. of CHI*, pages 779–788, 2009.

[19] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Selecting information diffusion models over social networks for behavioral analysis. In *Proc. of ECML PKDD*, pages 180–195, 2010.

[20] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Proc. of KES*, pages 67–75, 2008.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW*, pages 285–295, New York, NY, USA, 2001. ACM.

[22] X. Song, Y. Chi, K. Hino, and B. L. Tseng. Information flow modeling based on diffusion rate for prediction and ranking. In *Proc. of WWW*, pages 191–200, 2007.

[23] X. Song, B. L. Tseng, C.-Y. Lin, and M.-T. Sun. Personalized recommendation driven by information flow. In *Proc. of SIGIR*, pages 509–516, 2006.

[24] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *Proc. of KDD*, pages 807–816, 2009.

[25] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. of STOC*, pages 67–74, 2008.

[26] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proc. of WSDM*, pages 261–270, 2010.

# APPENDIX

# A. PROOF OF THEOREM 1

PROOF. It is sufficient to prove the property for a fixed number of time slots at $[K] = \{1, 2, \ldots, K\}$.

We use coupling technique to prove $\text{InfUser}^j(S_j)$ is monotone and submodular. Recall $S_j$ is the seed set for thread $j$, i.e., the set of users whose sidebars display thread $j$. We first provide the proof in the case that no user has been participating the thread at time slot 0. We show how to remove this assumption in the end of the proof. We build a directed graph $H$, which will encode all possible outcomes of the random events in our model. Figure 7 illustrates an example of the graph we will be constructing. In the graph $H$, there are $K + 2$ levels of nodes (recall that the discrete time slots that we are interested in in the process is $[K]$). In the first level, there is a virtual node $\tau$ and the users node set $\mathcal{U} = \{u_1, \ldots, u_n\}$. In total, there are $n + 1$ nodes in this level. In the second level, the node set is $\{u^1_{1,0}, u^2_{1,0}, u^1_{2,0}, u^2_{2,0}, \ldots, u^1_{n,0}, u^2_{n,0}\}$. In other words, each user $u_i$ has two correspondences in the second level. The node set is $\{u_{1,i-2}, u_{2,i-2}, \ldots, u_{n,i-2}\}$ for level $3 \le i \le K+2$.

Next, let us describe the edges of the graph $H$. Edges are all pointing from upper(smaller) levels to lower(larger levels. And there is no edge in the same level. From the first level to the second level, the edge set is as follows: there is an edge from $\tau$ to every node $u^1_{i,0}$ (for $i \in [n]$) in the second level. Each node $u_i$ (for $i \in [n]$) connects to the the node $u^2_{i,0}$. In other words, the edge set between the first and second level is $\{(\tau, u^1_{i,0}), (u_i, u^2_{i,0}) : i \in [n]\}$. Between the second and the third level, there are edges between $u^1_{i,0}$ and $u_{i,1}$ and between $u^2_{i,0}$ and $u_{i,1}$, i.e., the edge set between the second and third level is $\{(u^1_{i,0}, u_{i,1}), (u^2_{i,0}, u_{i,1}) : i \in [n]\}$. The set of edges starting from level 3 or below [7] is $\{(u_{j,t}, u_{i,t'}) : (i \neq j) \wedge t < t' \in [K]\}$. Finally, there is an edge from $\tau$ to every node on level 3 or below. i.e., the edges $\{(\tau, u_{i,t}) : i \in [n], t \in [K]\}$ are in $E(H)$.

All nodes are colored either black or white in the graph $H$. In our setting, colors are used to represent the outcomes of the event related to visiting and edges are used to represent event related to writing. Specifically, if the node $u_{i,t}$ is colored black, the user $i$ visits the thread on time slot $t$ for $t > 1$. The edges shall be interpreted as the possibilities that a user (the destination) in a later time slot will write a post because she is influenced by another user (the source) who wrote a post earlier (or influenced by the thread $\tau$). Details of defining the semantic of the graph $H$ is as follows.

**Coloring.** The nodes in the first level $\{\tau, u_1, \ldots, u_n\}$ in $H$ are colored black by default. The colors of the nodes below the third level represent the outcomes of the visiting events. In particular, for each node $u_{i,t}$ with $t > 1$ (the case for $t = 1$ will be discussed separately shortly), if user $u_i$ visits the thread at time $t$, we color it black, which is with probability $r_t$; otherwise, we color it white. Notice that for $t > 1$, whether a node is colored as black is independent of $S_j$, the set of seed users whose sidebars display the thread.

The set of nodes $u^1_{i,0}$ and $u^2_{i,0}$ in the second level and edges associated with them are designed to incorporate the increased visiting probability at time 1 by selecting the seeds $S_j$. Two Bernoulli random variables with parameters $\delta_1$ and

[7] To be clear, the $i$th level is below the 3rd level if $i > 3$.

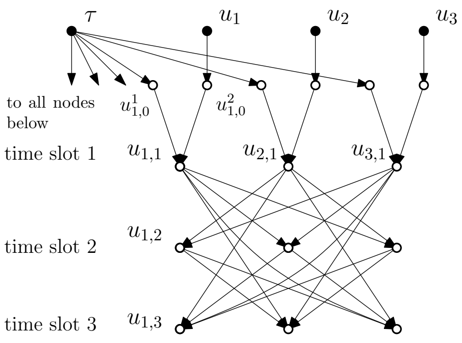

**Figure 7: The full graph $H$**

$\frac{\delta_0-\delta_1}{1-\delta_1}$ are used to simulate the visiting event with probability $\delta_0$ when the user $u_i$ is chosen as a seed and with probability $\delta_1$ otherwise. Specifically, the node $u^1_{i,0}$ is colored as black with probability $\delta_1$ and the node $u^2_{i,0}$ is colored as black with probability $\frac{\delta_0-\delta_1}{1-\delta_1}$. Also, the nodes $u_{i,1}$ ($i \in [n]$) on the third level are dummy nodes which are always colored in black. This level's nodes do not provide information regarding whether users will visit the thread on $t = 1$. Instead, a user $u_i$ will visit the thread on $t = 0$ if and only if: at least one of $u^1_{i,0}$ and $u^2_{i,0}$ is black when $u_i \in S_j$; and $u^1_{i,0}$ is black when $u_i \notin S_j$.

Finally, let us call the outcomes of all visiting events associated with all nodes $RR$.

**Subgraphs.** Next we define the writing events on edges. By removing a subset of edges in $H$ (and therefore retaining a subgraph of $H$), we encode the events a user decides to *not* write a post when she is being influenced by a second user.

Recall that in our model each user will visit existing posts only once, which implies an existing post has only one chance to influence a second user. As one user writes at most one post in the thread, each user only has one chance to influence another user. Hence, we can first fix all the outcomes of the influence probabilities, namely $RW$. There needs in total $\frac{n(n+3)}{2}$ random variables, including all pairwise events between users $u_i$ and $u_j$ for $i \neq j$ and $n$ writing events from the virtual user $\tau$.

We obtain a subgraph of $H$ based on $RW$ as follows. For $u_i \neq u_j \in \mathcal{U}$, if the outcome of the write event from $u_i$ to $u_j$ is false, we remove all edges from $u_{i,t}$ to $u_{j,t'}$ for all $t' > t$. For $u_i \in \mathcal{U}$, if the outcome of the write event from virtual user $\tau$ to $u_i$ is false, we remove all edges from $\tau$ to $u_{i,t}$ for $t \in [K]$ and $u^1_{i,0}$ and the edge from $u_i$ to $u^2_{i,0}$.

Let $H(RW, RR)$ be the subgraph and the coloring of the nodes obtained. $u_i$ posts in the thread, if and only if at least one of the *black* nodes in $\{u_{i,t}\}$ for $t \in [K]$ will be reachable from $\{\tau\} \cup S_j$ in $H(RW, RR)$ by a path of *black nodes.* Let $\mathrm{I}^i_{RW,RR}(S_j)$ be the indicator variable on whether $u_i$ posts with the outcome $RW$ and $RR$. We can write $\mathrm{InfUser}^j(S_j)$ as:

$$\mathrm{InfUser}^j(S_j) = \sum_{RW,RR} \Pr[RW]\Pr[RR] \sum_{i\in[n]} \mathrm{I}^i_{RW,RR}(S_j),$$

where $\Pr[RW]$ and $\Pr[RR]$ are the probabilities to have the outcomes $RW$ and $RR$ respectively.

Therefore, it is sufficient to prove that $\mathrm{I}^i_{RW,RR}(S_j)$ is monotone and submodular. To show that, we merge all *black* nodes $\{u_{i,t}\}$ for $t \in [K]$ to a super node $\bar{u}_i$ by redirecting all edges originally to these black nodes to the new node $\bar{u}_i$. Then we remove all white nodes from the graph. Afterwards, $\mathrm{I}^i_{RW,RR}(S_j)$ is simply the indicator variable on whether $\bar{u}_i$ is reachable from set $\{\tau\} \cup S_j$ in the new graph, which is clearly monotone and submodular.

Now consider the case that some nodes have been appeared in the thread before time slot 0. We simply remove those nodes in the counting of reachable node, which will not affect the submodularity property. □

# B. PROOF OF THEOREM 2

Proof. We first show that the following *maximum overlapping set* problem is equivalent to a special case of the participation maximization problem. An instance of the maximum overlapping set problem is a special class of directed graphs, in which vertices are divided into two sets $U = \{u_1, u_2, \ldots, u_m\}$ and $V = \{v_1, v_2, \ldots, v_n\}$, and all directed edges are from nodes in $U$ to nodes in $V$, and every $v \in V$ is incident to some edges. For some $S \subseteq U$, Let $C(S)$ be the subset of $V$ that is covered by $S$, that is, the set of vertices in $V$ having neighbors in $U$. The problem is to find a partition of $U$ into two sets $U_1$ and $U_2$, such that the overlap of their coverage in $V$,$C(U_1) \cap C(U_2)$, is the largest.

To see that it is equivalent to a special case of our participation maximization problem, we take any above graph $G$ as our social network, where all edges in $G$ have weights 1. For the virtual user $\tau$, all edges from $\tau$ into $U$ have weight 1 while all edges from $\tau$ to $V$ have weight 0. We only need two threads, with sidebar size $B = 1$. The normal visit probabilities $\delta_i$ are all 0 except for the second time slot $\delta_2 = 0.5$. The boosted visit probability $\delta^* = 1$. We want to allocate threads to sidebars in the first time slot ($s = 1$).

Under the above setting, sidebar allocations partition all users in $U$ into $U_1$ and $U_2$, corresponding to the two threads (it also partitions $V$, but it does not matter for our purpose). Consider the first thread, which is shown in the sidebars of users in $U_1$ in the first time slot. For every user in $U_1$, in slot 1 she visits the thread (since $\delta^* = 1$), influenced by $\tau$ and writes a post. All other users do not visit the thread since $\delta_1 = 0$). In the second time slot, every user in $C(U_1)$ has a probability $\delta_2 = 0.5$ to visit the thread, and once in she will be influenced by some users in $U_1$ who already posted and write a post. For every user in $U_2$, she also has a probability 0.5 to visit the thread and then influenced by $\tau$ to write a post. For every user in $V \setminus C(U_1)$, she may visit the thread, but even if so, she will not be influenced by $\tau$ nor by any user in $U_1$, and thus she will not write a post in the thread. No user will visit any thread after slot 2. Therefore, summing up all, the expected number of users who post in thread 1 is $|U_1|+0.5|U_2|+0.5|C(U_1)|$. Symmetrically, the expected number of users who post in thread 2 is $|U_2| + 0.5|U_1| + 0.5|C(U_2)|$. Therefore, the expected number of total participants is $1.5(|U_1| + |U_2|) + 0.5(|C(U_1)| + |C(U_2)|) = 1.5|U| + 0.5|V| + 0.5|C(U_1) \cap C(U_2)|$. As the result, maximizing this value is equivalent to maximize the intersaction $C(U_1) \cap C(U_2)$.

We now show that the maximum overlapping set problem is NP-hard, by a simple reduction from the MaxCut problem [7]. Given an undirected graph $G = (V, E)$, MaxCut problem is to find a cut of maximum size. We convert $G$

into $G' = (V \cup E, E')$ where $E' = \{(v, e) \mid v \in V, e \in E\}$. Then the overlap between $C(U_1)$ and $C(U_2)$ for a partition $U_1, U_2$ of $U$ is exactly a cut of $G$ into $U_1$ and $U_2$. Thus a solution to the maximum overlapping set problem of this instance provides a solution to MaxCut. □

# C. ALGORITHM TO LEARN INFLUENCE PROBABILITIES

## C.1 Variables

Let $P(T_j)$ be the post sequence of $T_j \in \mathcal{T}$, where $P(T_j) = (\langle \tau, t_0 \rangle, \langle u_1, t_1 \rangle, \langle u_2, t_2 \rangle, \ldots, \langle u_\ell, t_\ell \rangle)$, indicating $u_i$ posts to $T_j$ at time $t_i$, and superscript $j$ is omitted for convenience. The corresponding thread rank at time $t_i$ is $r_i$ and the visit probability is $\delta_{r_i}$. We define the following variables for notational convenience of our iterative algorithm:

- $\sigma^j_{i,v}$: the probability that user $v$ visits $T_j$ within $[t_i, t_{i+1})$. If $i = l$, i.e., the post is the last one, since there is no subsequent post, the time window is $[t_l, t_l + \Delta t)$.
- $\mathcal{V}^j_v(k, i)$: the probability that user $v$ visits $T_j$ within $[t_k, t_{k+1})$ and $[t_i, t_{i+1})$ respectively, and $v$ does not visit $T_j$ during time $[t_{k+1}, t_i)$.
- $\mathcal{I}^j_v(k, i)$: the probability that user $v$ is influenced by at least one of the preceding users and becomes the $(i+1)$th post writer in $T_j$, besides, the previous visit is within $[t_k, t_{k+1})$.
- $\mathcal{N}^j_v(i)$: the probability that user $v$ visits $T_j$ within $[t_i, t_{i+1})$, does not post and never revisits afterwards.

For *online* users, in every time period $[t_i, t_{i+1})$, we calculate the average value of all visit probabilities varying with thread ranks, as approximation for $\sigma^j_{i,v}$. For example, suppose user $v$ is online, and the first post of $T_j$ has thread rank $r_1 = 1$, while the second post has rank $r_2 = 3$. Then, during time period $[t_1, t_2)$, we assume that $v$ equally likely comes to the forum when the rank of the thread is 1, 2, or 3. Recall $\delta_r$ denotes the visit probability when thread rank value is $r$. Thus, all the possible visit probabilities are $\delta_1, \delta_2$ and $\delta_3$, and $\sigma^j_{1,v}$ is $(\delta_1 + \delta_2 + \delta_3)/3$. Formally, let Online$(v)$ be the union of time intervals in which $v$ is online.

$$\sigma^j_{i,v} = \begin{cases} \dfrac{\sum_{r_i \le r \le r_{i+1}} \delta_r}{r_{i+1} - r_i + 1}, [t_i, t_{i+1}) \cap \text{Online}(v) \neq \emptyset \\ \qquad 0, \text{ otherwise} \end{cases} \tag{3}$$

Note that all the $\sigma^j_{i,v}$ are independent but not mutual, because revisiting is allowed in our model. By Equation 3, $\sigma^j_{i,v}$ is 0 if $v$ is not online during $[t_i, t_{i+1})$. For notational convenience, we define $\delta^j_{-1,v} = 1$.

$\mathcal{V}^j_v(k, i)$ is the probability that $v$ visits $T_j$ within $[t_k, t_{k+1})$, does not visit during $[t_{k+1}, t_i)$, and re-visits in time period $[t_i, t_{i+1})$. $\mathcal{V}^j_v(-1, i)$ denotes that the visit in $[t_i, t_{i+1})$ is the *first* time $v$ visits $T_j$. Hence,

$$\mathcal{V}^j_v(k, i) = \sigma^j_{k,v} \sigma^j_{i,v} \prod_{k<x<i} (1 - \delta^j_{x,v}), k \in [-1, i) \tag{4}$$

$\mathcal{I}^j_v(k, i)$ denotes the following probability: user $v$ visit $T_j$ in $[t_k, t_{k+1})$ and all the post writers $\{u_x \mid 0 \le x \le k\}$ preceding $u_k$ fail to influence $v$. And user $v$ visit $T_j$ again in $[t_i, t_{i+1})$ without visiting $T_j$ during $[t_{k+1}, t_i)$. Furtheremore, at least one of the post writer $u_x$ between $u_k$ and $u_i$, $\{u_x | k < x \le i\}$, successfully influences $v$ to post. Hence,

$$\mathcal{I}^j_v(k, i) = \mathcal{V}^j_v(k, i) (\prod_{0 \le x \le k} (1 - w_{u_x,v}))(1 - \prod_{k<x\le i} (1 - w_{u_x,v}))$$

$\mathcal{N}^j_v(i)$ is the probability that $v$ visits $T_j$ in $[t_i, t_{i+1})$ (with probability $\sigma^j_{i,v}$), not influenced by any preceding users (with probability $\prod_{0 \le x \le i}(1 - w_{u_x,v})$), and never revisits (with probability $\prod_{i<k\le l}(1 - \delta^j_{k,v})$). Thus,

$$\mathcal{N}^j_v(i) = \sigma^j_{i,v} \prod_{0 \le x \le i} (1 - w_{u_x,v}) \prod_{i<k\le l} (1 - \sigma^j_{k,v}) \tag{5}$$

## C.2 Learning $w_{u,v}$

With the above variables, following the user posting model in Figure 1, we design an iterative method to estimate the influence probabilities in $G_\tau$.

Incorporated with visit probability, we adapt the algorithm of [9, 20] for our forum user posting model. The algorithm iterates between two conditional probabilities: In threads that $v$ posts after $u$, step 1 computes the conditional probability that $v$ posts because of $u$'s influence given $v$ posts in $T_j$. While step 2 updates influence probabilities $w_{u,v}$ by estimating the probability that $v$ is influenced by $u$ given $v$ visits $u$'s post. In particular, we estimate $w_{u,v}$ with the number of times $v$ is influenced by $u$ divided by the number of times $v$ reads $u$'s post.

**Step 1:**

$$p^j_{u,v} = \frac{w_{u,v}(\prod\limits_{k<\lambda^j_u} (1 - w_{u_k,v})) \sum\limits_{-1 \le k < \lambda^j_u} \mathcal{V}^j_v(k, \lambda^j_v - 1)}{\sum\limits_{-1 \le k < \lambda^j_v - 1} \mathcal{I}^j_v(k, \lambda^j_v - 1)}$$

In step 1, let $p^j_{u,v}$ be the conditional probability that $u$ influences $v$ to post in $T_j$ given $v$ posts in $T_j$ in the particular time stamp. For brevity of equations, suppose in $T_j$, $u$ writes the $\lambda^j_u$-th post and $v$ writes the $\lambda^j_v$-th post, $\lambda^j_v > \lambda^j_u$. For other threads, we set $p^j_{u,v} = 0$.

In the numerator,, Pr($u$ influences $v$) is: $u$ successfully influences $v$ (with probability $w_{u,v}$), and all the users preceding $u$ in $T_j$ failed (with probability $(\prod_{k<\lambda^j_u}(1 - w_{u_k,v}))$. Besides, $v$ visits within $[t_{\lambda^j_v - 1}, t_{\lambda^j_v})$ and the last visit occurs before $\lambda^j_u$, with probability $\sum_{-1 \le k < \lambda^j_u} \mathcal{V}^j_v(k, \lambda^j_v - 1)$. Because if the last visit occur after $u$, it implies $u$ fails to influence $v$. Recall we only record the first post, and revisiting *posted* threads is ignored in the user model.

In the denominator, recall that $\mathcal{I}^j_v(k, i)$ denotes the probability that user $v$ becomes the $(i+1)$th post writer in $T_j$, besides, the previous visit is within $[t_k, t_{k+1})$ for $-1 \le k$. And Pr($v$ posts in $T_j$) is the probability that $u_i$ posts in $T_j$ at position $i$, $p^j(u_i)$, which is $\sum_{-1 \le k < \lambda^j_v - 1} \mathcal{I}^j_v(k, \lambda^j_v - 1)$.

**Step 2:**

$$w_{u,v} = \frac{\sum\limits_{j \in S} p^j_{u,v}}{\sum\limits_{j \in S} \sum\limits_{\lambda^j_u \le k < \lambda^j_v} p^j_{u_k,v} + \sum\limits_{j \in S'} \dfrac{\sum_{\lambda^j_u \le k \le l} \mathcal{N}^j_v(k)}{\sum_{-1 \le k \le l} \mathcal{N}^j_v(k)}}$$

In step 2, let $S$ denote the set of threads that $v$ posts after $u$'s post and let $S'$ denote the set of threads that $u$ posts but $v$ does not.

The influence probability $w_{u,v}$ is estimated as: the number of times $v$ actually writes due to the influence from $u$ divided by the number of times $v$ might read $u$'s post. The latter should be further divided into two parts: the first part is the set of threads $S$, i.e., $v$ posts after $u$; the second is the set of threads $S'$, i.e., $u$ posts but $v$ does not. For one thread in $S$, $v$ reads $u$'s post if and only if $u$ or posts after $u$ triggers $v$. That is because, in our user posting model, if $v$ is triggered by posts before $u$, she will write and leave without reading $u$'s post. Hence, the probability that $v$ reads $u$ in thread $T_j \in S$ is: $\sum_{\lambda_u^j \leq k < \lambda_v^j} p_{u_k,v}^j$.

For the second part of threads $S'$, it is the conditional probability that $v$ reads $u$, given $v$ does not post in $T_j$. Recall that $\mathcal{N}_v^j(i)$ denotes the probability that the last time that $v$ visits $T_j$ is after $u_i$ in $[t_i, t_{i+1})$, does not post and never revisits. Only when the last visit is after $u$, $v$ has a chance to read $u$. Thus, $\Pr(v$ reads $u$ without posting in $T_j)$ $=\sum_{\lambda_u^j \leq k \leq l} \mathcal{N}_v^j(k)$. $\Pr(v$ does not post in $T_j)$ is the probability that $v$ does not post in $T_j$, which is $\sum_{-1 \leq k \leq l} \mathcal{N}_v^j(k)$. Therefore, the probability that $v$ actually reads $u$ in $T_j$ is $\frac{\sum_{\lambda_u^j \leq k \leq l} \mathcal{N}_v^j(k)}{\sum_{-1 \leq k \leq l} \mathcal{N}_v^j(k)}$.

We can assign a set of initial values of $w_{u,v}$. Then we directly obtain an iterative algorithm by applying the two steps above until the solution of $w_{u,v}$ converges. This algorithm is very simple and always converge to a unique solution regardless of initial values of $w_{u,v}$ in our study.