Participation Maximization Based on Social Influence in Online Discussion Forums *

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, 5F Beijing Sigma Center. {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 the *overall* forum participation. To this end, we propose a mechanism to maximize user influence and boost participation by displaying forum threads to users. We formally define the *participation maximization* problem, and 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 is impracticable for real-world forums due to time complexity. 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 in maximizing participation.

The results of this work demonstrates that current recommender systems can be made more effective by considering future influence propagations. The problem of participation maximization based on influence also opens a new direction in the study of social influence.

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 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 values associated with the online forums, the owners of the forums also directly benefit from the traffic of active forums, e.g. more traffic means more advertising revenue.

Being able to build an *active* online forum platform that encourages users to participate in discussions would also be beneficial to individual users. When a user submits a new thread (a new thread means one user creates an initial post to start a new discussion in a forum), besides accurate answers or valuable suggestions, he also hopes for an active 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 blog users care about the number of comments.

To this date, albeit the considerable progress in system design to enable 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 increase user participation into online discussions. However, there have been several successful Q&A services that leverage information from social networking profiles, such as Aardvark (Horowitz and Kamvar 2010) (http://vark.com/) and Quora (http://quora.com/) that connect to Facebook or Twitter networks. Facebook itself also rolled out an ambitious Q&A service "Questions" in July 2010, which has been billed as Killer App considering its resource of 600 million users. The success of the above services revealed that social ties have a positive effect on users' participation.

Although there are typically no explicit social ties (i.e. friendship in Facebook) in online discussion forums, we observe that users tend to post after certain users — the effect of social influence. 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 recommendation. In recommender 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. 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 a forum as a whole. Hence, besides predicting which threads users will be most interested in through historical data, we should further look into the future to maximize the forthcoming influence diffusion.

Existing models in maximizing influence diffusion that market to the influencers only (Kempe, Kleinberg, and Èva

*The work was completed when Tao, Zhenming and Xiaorui were doing internships in Microsoft Research Asia.

Tardos 2003; Chen, Wang, and Yang 2009) also do not fit into our scenario. For example, suppose that we identified the influencers and recklessly encouraged them to participate in every thread, they would feel disturbed and find the recommendation unhelpful. Moreover, everybody is an influencer in some way. Thus, for practicability and the benefit of the social media ecosystem, a small number of threads should be allocated to every user. This leads to a new formulation — the *participation maximization* problem — an optimal allocation problem to maximize overall participation in a forum through influence propagation.

More specifically, with the purpose of maximizing the total participants, each online users will be displayed B threads, to increase the chance of his participation as well as the subsequent influence propagation to more users. 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 (Dobzinski and Schapira 2006; Vondrák 2008). For efficiency, we further propose a heuristic algorithm, 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. Through comparison with other algorithms including a personalized recommendation algorithm (Song, Tseng, and et al. 2006) and a social welfare maximization algorithm (Dobzinski and Schapira 2006) on data from a real forum, we show that TABI performs consistently as the most effective algorithm in maximizing total participation.

To summarize, our main contributions are as follows:

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

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

(iii) We suggest that when making recommendations, besides predicting users' interests based on historical data, considering the future influence propagation is also important for the overall forum participation.

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 briefly discuss two relevant areas of works as below.

Learning social influence in the social network. An important task in the study of social influence is to learn the strength of social influence among users. Gruhl et al. (Gruhl, Guha, and et al. 2004) used a variant of independent cascade model in blogspere and informally derived an Expectation-maximization(EM)-like algorithm to induce the influence probabilities among users. Saito et al. (Saito, Nakano, and Kimura 2008) derived a similar E-M algorithm in a more formal analysis to estimate influence probabilities. Goyal et al. (Goyal, Bonchi, and Lakshmanan 2010) tackled the same problem in another variant of the influence propagation model, and applied Maximum Likelihood Estimator (MLE). Influence learning provides the social 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 (Saito, Nakano, and Kimura 2008) to extract social influence in TripAdvisor, and use it as input to our participation maximization algorithm.

Applications of social influence in social media. Extensive studies have been conducted to apply social influence in viral marketing (Kempe, Kleinberg, and Èva Tardos 2003; 2005; Chen, Wang, and Yang 2009; Chen, Wang, and Wang 2010), personalized recommendation (Song, Tseng, and et al. 2006), ranking (Weng, Lim, and et al. 2010), etc. Very recently, in (Ienco, Bonchi, and Castillo 2010), a similar problem was studied independently—how to maximize the activity of Microblogging network by showing each user k memes. Compared to their work, we provide more detailed proof in problem formulation and formulate it to a special instance of Social Welfare Maximization problem. Moreover, we compare our result to recommendation methods.

3 User Posting Model Based on Influence

In this section, we describe our model of user posting behavior in online discussion forums based on social influence. 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 T 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* τ, together with edges from τ 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 weight $w_{\tau,u}$ for each edge (τ, u) with $u \in \mathcal{U}$. Intuitively, the virtual user τ represents the content of the threads, and $w_{\tau,u}$ represents how the content of the threads affect users' posting behaviors. Note here $\mathcal{F}$ indicates one forum on a specific topic, more specifically, one category in TripAdvisor, U indicates users who participate in $\mathcal{F}$ and $\mathcal{T}$ of $\mathcal{F}$ is a group of threads with the specific topic (a.k.a. all threads in one category). Thus we only introduce one virtual user for one $\mathcal{F}$, without adding different virtual users per thread.

Figure 1 shows the diagram of the user posting model. For

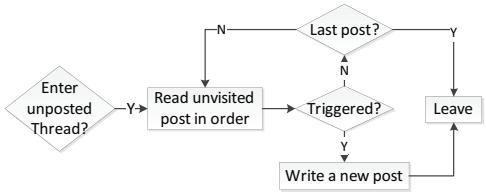

Figure 1: Diagram on the user posting model

an online user v, once v visits a thread $T \in \mathcal{T}$, v will start reading unvisited posts in order. When v reads the post by u, v is influenced by u to write a post in this thread T with probability $w_{u,v}$. 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) all users have read all the existing posts in the thread but are not influenced to write one; or (b) all users have posted in the thread.

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 (Kempe, Kleinberg, and Èva Tardos 2003). 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 (Gruhl, Guha, and et al. 2004; Anagnostopoulos, Kumar, and Mahdian 2008; Tang, Sun, and et al. 2009; Saito, Kimura, and et al. 2010; Goyal, Bonchi, and Lakshmanan 2010). 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.

Topic differentiation. In our model, we treat threads in one $\mathcal{F}$ (category) equally important. One may argue that some threads are more popular. We could further categorize threads by topic model or level of quality to obtain different $w_{\tau,v}$. But the further categorization can be viewed as dividing one $\mathcal{F}$ into a set of sub-categories $\{\mathcal{F}'\}$. Recall that for one $\mathcal{F}$, we allocate threads $\mathcal{T}$ of $\mathcal{F}$ to users U who participate in $\mathcal{F}$. Hence, we do not further differentiate topics for simplicity and clarity.

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. This simplification can be justified as follows. First, our optimization object is to maximize the number of distinct participants, not the number of posts generated, and thus multiple posts by a single user do not directly affect. 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. 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.

4 Participation Maximization

We propose a novel use of the sidebar mechanism based on influence propagation to increase user participation in online discussion forums. Sidebar is used as an example to illustrate our mechanism, other user interfaces such as pop-up list could also be adopted. 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 has the submodularity property, making the problem as an instance of social welfare maximization with submodular functions.

For convenience, we discretize continuous time into time slots denoted as slot $1, 2$ and so on. Threads added into $\mathcal{F}$ at different time slots, which share the same optimization function as presently shown in Equation (1), are treated as different instances for the optimization purpose. Hence, in the following, we take threads generated in one slot to exemplify our approach.

4.1 Problem formulation

We define the *participation maximization* problem as follows. Each user has a budget constraint sidebar to display B threads, where B is a small constant (usually 5 or 10). With a scheme that optimizes the total participation among all the threads, at a certain time slot s, the system allocates B different threads to each user, so that the user would visit threads in his sidebar with a higher probability δ^*, compared to the original probability δ. We use only one time slot for the allocations, and threads initiated at other time slots can be allocated at other time slots with the identical mechanism.

According to our user posting model as shown in Figure 1, because visit probabilities to the threads displayed in sidebars are boosted, the mechanism can increase the probability that users posts in their suggested threads in succession. In turn, these posts may further influence subsequent users and increase the probability that others write posts in the threads. Thus, the overall number of participants (those who write posts) in $\mathcal{F}$ is increased.

Formally, let $S_j \subseteq \mathcal{U}$ be the set of users whose sidebars display thread T_j, and $\mathrm{InfUser}^j(S_j)$ 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 (1) the social influence graph G_τ, (2) a sequence of visit probabilities δ_j's, (3) thread set $\mathcal{T}$, (4) time slot $s \geq 1$ for sidebar allocation, (5) prefix of posts sequences up to slot $s-1$, (6) sidebar size B, (7) boosted visit probability δ^*, the problem of participation maximization is to find a partition $\{S_1, S_2, \ldots S_m\}$ of $\mathcal{MU}$ which maximizes the total (expected) number of participants in all threads as

$$\sum_{j=1}^{m} \mathrm{InfUser}^j(S_j) \tag{1}$$

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 (Kempe, Kleinberg, and Èva Tardos 2003). 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 events of each node are encoded by a random coloring process. Then the influence function is simply to count the number of reachable nodes with a particular color from a seed set, which is clearly submodular. Due to space constraint, the complete proof is included in our full technical report (Sun, Chen, and et al. 2010). □

4.3 Comparison with related problems

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

Comparison with recommender systems. In the context of online discussion forums, techniques in recommender systems (Song, Tseng, and et al. 2006; Sarwar, Karypis, and et al. 2001) can certainly be used to assign threads to sidebars of interested users and potentially increase their participation. However, in the 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 potentially will 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 recommendations. Considering the future influence generated by the current recommendations is the novelty differentiating our work from other recommender systems.

Comparison with influence maximization for viral marketing. The influence maximization problem is to find a small seed set in a social network to maximize their eventual influence spread (Kempe, Kleinberg, and Èva Tardos 2003; 2005; Chen, Wang, and Yang 2009). By comparison, we aim at maximizing the total participation among all the threads, *not* participants in a specific thread; and the constraint is on the number of threads each user can be recommended, *not* on the number of users each thread can be recommended to. Hence, the problem formulation becomes markedly unlike influence maximization, and thus requires different solutions.

Comparison with social welfare maximization. In social welfare maximization problems (Dobzinski and Schapira 2006; Vondrák 2008), 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 $\mathrm{InfUser}^j(S_j)$ with submodular property. Therefore, participation maximization is a specific instance of social welfare maximization with submodular utility functions.

5 Thread Allocation Algorithms

In this section, we discuss algorithms to allocate threads, and propose a 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.

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 Max-Cut problem. The complete proof is included in our technical report (Sun, Chen, and et al. 2010). □

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, 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 τ, and thus the utility functions $\mathrm{InfUser}^j(\cdot)$ are same for all threads, in which case Vondrák (Vondrák 2008) 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 $s = 1$.

But when $s \geq 2$, most threads already have some posts (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 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 (Dobzinski and Schapira 2006). 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 (Dobzinski and Schapira 2006; Vondrák 2008) can be applied to solve the participation maximization problem. Algorithm 1 presents our adaptation of a $(2 - \frac{1}{m})$-approximation algorithm (Dobzinski and Schapira 2006), where m is the number of threads in our model. Essentially, the algorithm computes the incremental effect R_j of assigning thread T_j to

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: select exactly one thread j randomly as follows: each thread j is chosen with probability $\frac{R_j^{m-1}}{\sum_{T_k \in \mathcal{T}} R_k^{m-1}}$
7: update $S_j = S_j \cup \{v\}$ and $P_v = P_v - 1$.

Algorithm 2 TABI

1: **for** each $v \in \mathcal{U}$ **do**
2: **for** each $j \in \mathcal{T}$ **do**
3: calculate ΔInf_v^j as Equation 2
4: Rank threads by ΔInf_v^j in descending order
5: Select top B threads to display in v's sidebar

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 6). 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 he is online, independent of assignments of users who log in later.

However, RPA as well as other approximation algorithms assumes that the computation of utility function is done by an oracle. Thus RPA is infeasible in real forums, because it requires sufficient amount of simulations to estimate $\text{InfUser}^j(S)$. Our experimental results in the next section show that RPA is extremely time consuming and preforms poor under insufficient number of simulations. This leads us to consider fast heuristic algorithms.

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_τ, 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 ΔInf_v^j that brought by displaying thread T_j to v is estimated as:

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

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 δ^* 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 ΔInf_v^j by δ^* for ranking and selection.

The above estimate contains two parts: *(i)* the first parenthesis, which captures how likely 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 recommender system, while the second part focuses on incorporating future influence into thread selection, which we believe is our unique consideration differing from recommender 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.

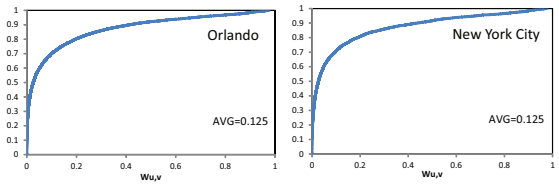

Figure 2: CDF of influence probabilities

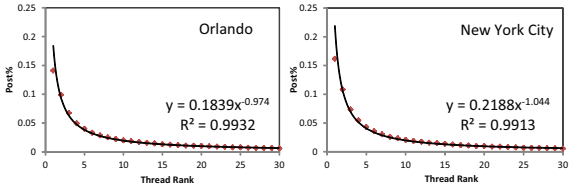

Figure 3: Post distribution v.s. thread rank

6 Experiments

In this section, we use data from a real-world online discussion forum to evaluate the effectiveness of TABI and compare it against several other algorithms. We first extract parameters, such as the influence network and visit probabilities, from the forum data as inputs for algorithms, and then compare the expected number of participants via simulation.

6.1 Datasets

Our datasets are crawled from TripAdvisor's World travel forum, which represents the largest travel community in the world. The 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 popular categories: Orlando, London and New York City (NYC).

Even though we have crawled data for several years, most users have a short active period on social media (Guo, Tan, and et al. 2009). The influence among users are also likely to change over time. Thus we choose a window size *t_win* to ensure around 80% of users have their forum life spans (the period between his first and last post) within *t_win*. In TripAdvisor, *t_win* is about 60 days. Thus, we choose a 60 day

period from 01/01/2009 to 03/01/2009 for our experiments. Within the period, in category Orlando, London and NYC, there are $4062, 1800, 2455$ threads and $2085, 1467, 1694$ distinct users, respectively.

6.2 Extracting parameters

The Social Influence Network. In the formulation of the participation maximization problem (Section 4), the social influence network is treated as an input of the problem. But no explicit social relationships are maintained in TripAdvisor, so we need to construct an implicit influence network and learn the influence probabilities on the network.

Intuitively, if one user's post influences another user and lead to his 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 (Gruhl, Guha, and et al. 2004; Saito, Nakano, and Kimura 2008; Goyal, Bonchi, and Lakshmanan 2010; Saito, Kimura, and et al. 2010). Based on our forum context, we adapt the E-M algorithms in (Gruhl, Guha, and et al. 2004; Saito, Nakano, and Kimura 2008) 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, the detailed learning algorithm is given in our technical report (Sun, Chen, and et al. 2010). The Cumulative Distribution Functions (CDF) are given as Figure 2 (London with similar distribution is omitted due to limited space).

Visit probabilities. Since TABI (Algorithm 2) does not depend on visit probabilities, we would like to test the algorithm against different visit probability sequences. Meanwhile, we want to obtain a visit probability sequence that is similar at least in trend to the real data. However, accurate estimation of visit probabilities is impossibile due to the lack of login and browsing data of TripAdvisor users. Therefore, we make estimation from the crawled posting data.

Following previous studies on visit probability (Hogg and Szabo 2009; Ienco, Bonchi, and Castillo 2010), we get the estimation based on *recency*. More specifically, we define *thread rank r* of thread $T \in \mathcal{T}$ at a time t as: its rank in the reversed chronological order of all threads at t. For example, among all m threads at t, the latest (submitted most recently) thread has $r = 1$, while the oldest has $r = m$. Every time there is a post in T, the post can be assigned with T's thread rank value r. Then the visit probability δ_r for threads with rank r, is proportional to the ratio between the number of posts with r and the total number of posts among all threads, as shown in Figure 3. Both curves of Orlando and NYC fit very well into power-law curves, with power-law exponents α 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 more attention to recent threads than earlier threads but there is always users visiting old threads.

Algorithm 3 Simulate Existing Participants

Input: visit probability sequence δ_t, influence network G_τ, time slot s for thread allocation
Output: existing participants in each thread j, denoted as EP_j
Initialize $EP_j = \{\tau\}$ for each j
for time slot $t = 1$ to $s - 1$ **do**
  for each $v \in \mathcal{U}$ **do**
    for each $j \in \mathcal{T}$ with $v \notin EP_j$ **do**
      if v visits j with probability δ_t **then**
        for each $u \in EP_j$ **do**
          if v hasn't read u's post **and** is influenced with probability $w_{u,v}$ **then**
            $EP_j = EP_j \cup \{v\}$
            break

Algorithm 4 Simulate New Participants

Input: visit probability sequence δ_t, boosted visit pr δ^*, influence network G_τ, time slot s for thread allocation, maximum slot K, existing participants EP_j in each j, sidebar size B
Output: the number of new participants ($newParticipant$)
Allocate B threads to each v by one algorithm, so that each j is displayed to a set of users S_j
for time slot $t = s$ to K **do**
  for each $v \in \mathcal{U}$ **do**
    for each $j \in \mathcal{T}$ with $v \notin EP_j$ **do**
      if $v \in S_j$ **and** $t == s$ **then**
        $\delta_t = \delta^*$
      if v visits j with probability δ_t **then**
        for each $u \in EP_j$ **do**
          if v hasn't read u's post **and** is influenced with probability $w_{u,v}$ **then**
            $newParticipant++$
            $EP_j = EP_j \cup \{v\}$
            break

6.3 Simulation tests and results

Since we have not deployed our mechanism in a real online forum environment, we demonstrate its effectiveness via simulations based on the user posting model (Figure 1) and the analyzed parameters (influence network and visit probabilities). 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. We compare the following five algorithms:

1) NoSidebar, as the baseline;

2) Random, allocation at uniformly random;

3) RPA, as described in Algorithm 1;

4) TEABIF, a personalized recommendation algorithm, named topic-sensitive early adoption based information flow (TEABIF) (Song, Tseng, and et al. 2006), which recommends items to users by estimating whom the information will propagate to with high probabilities.

5) TABI, as described in Algorithm 2.

To simulate the process of participation, first, we generate

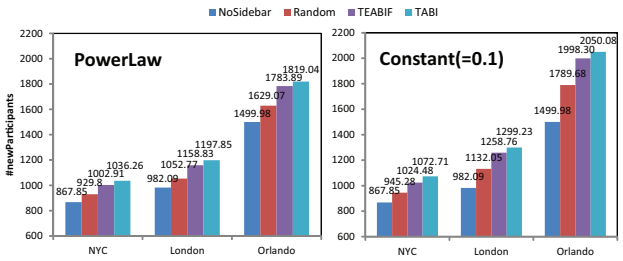

Figure 4: Results on different visit probability sequences

one *group* of $\{EP_j\}$ by Algorithm 3 to get existing participants. Then, we run Algorithm 4 for 1000 times to obtain an average number as new participants ($newParticipant$) for this group. We simulate 500 such groups and take the average number of $newParticipant$ as the final reported result. RPA would be extremely slow if we also run 1000 simulations to obtain one $\mathrm{InfUser}^j(S)$ value in Algorithm 1. To finish RPA in a reasonable amount of time, we run 10 simulations to estimate $\mathrm{InfUser}^j(S)$. 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.

Comparing the effectiveness of different algorithms. In the first test, we compare the effectiveness among the above five algorithms when thread number $m(=|\mathcal{T}|)$ is 30, 40, and 50. We set sidebar size $B = 5$, the slot for allocation $s = 2$, maximum time slot $K = 15$, and use δ_r described in Section 6.2 to approximate the visit probability δ_t. The boosted visit probability is set as $\delta^* = 0.8$. The value of δ^* would not affect thread allocation of Random, TEABIF and TABI, and thus total participation has a linear relationship with δ^*. For RPA, its thread allocation depends on δ^* when calculating $\mathrm{InfUser}^j(S)$, but our simulation results show that total participation is still close to a linear relationship with δ^*. Therefore, results for other δ^* values only have a constant factor difference and can be derived, so we do not report the exact numbers here.

The results of category NYC, London and Orlando are given in Figure 5. In all nine tests covering three categories and three different numbers of threads m, TABI 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 ± 6.32,20.13 ± 6.51,27.52 ± 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. As mentioned above, TABI does not depend on visit probabilities. In the second test, we intend to verify that TABI could perform consistently better than other algorithms under different visit probability sequences. To do so, we remain all the parameter settings in the first test except that replacing δ_t 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 δ_r.

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

Figure 4 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 all other methods are consistent.

Effectiveness on different allocation time slots. In the third test, we aim at checking whether TABI could perform consistently better than other algorithms under different allocation slot s. To this end, we vary s from 2 to 10 ($s = 1$ can be solved by random allocation), set $m = 40$ and a different $\delta^* = 0.5$. Since different allocation time slots have different groups of pre-existing participants, in order to compare the results of different allocation slots in a fair way, we use the number of participants, instead of the *additional* number of participants ($newParticipant$) as metric. It is computed as $|\cup_j EP_j|$ after running Algorithms 3 and 4.

Our results show that TABI outperforms TEABIF and Random in all the allocation time slot s, which proves 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 poor user experiences. Therefore, we believe a better conclusion is that larger boost in visit probabilities may provide more participation, but the selection of time slot s for allocation should consider other factors such as user experiences. This is why we use s as a parameter of the problem rather than a variable to be tuned empirically.

Summarization Our simulation results demonstrate that sidebar mechanism based on social influence can significantly improve participation, and TABI outperforms the four aforementioned algorithms, including an approximation algorithm and a personalized recommendation algorithm. We believe that the reason of the better performance of TABI, especially when comparing with recommender systems such as TEABIF, is because TABI considers social influence that may increase future participation.

Although simulation-based evaluation provides valuable insights to the understanding of the algorithms, it certainly has its limitations. Our simulation is based on a simplified

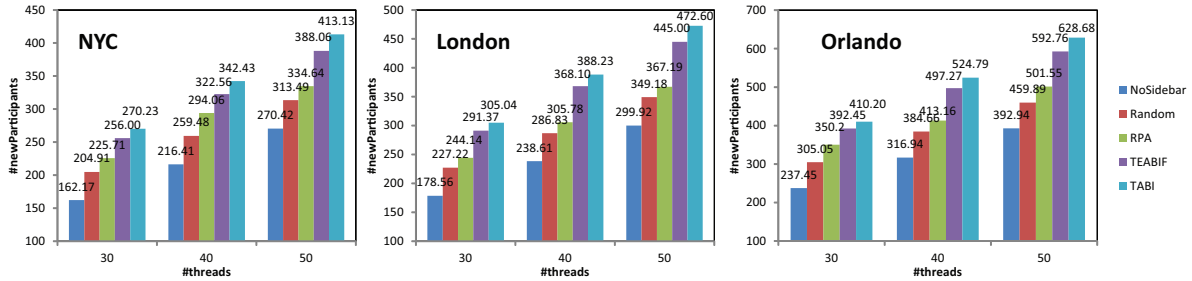

Figure 5: Results of Five Approaches

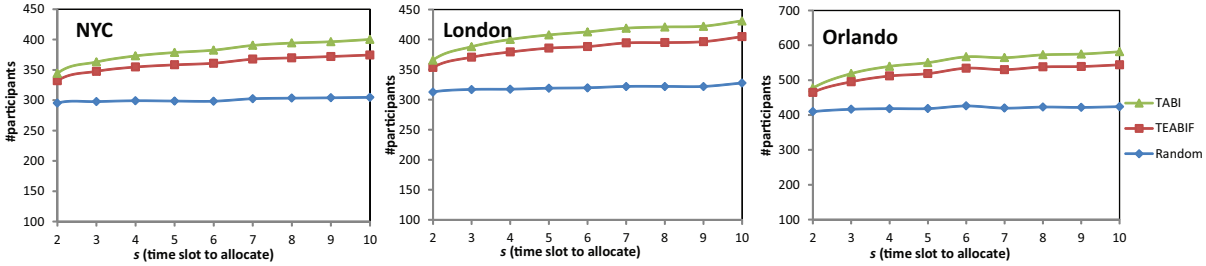

Figure 6: Results on different allocation time slots

user model, which does not cover several effects in the real world, such as off-topic posts, login frequency, patience to read existing posts, etc. To overcome the limitations, we have to further enrich the user model, and rely on user studies for model validation, which will be our future directions.

7 Conclusion

To summarize, in this paper, we propose a personalized allocation mechanism to maximize total 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 with 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, and validate the robustness and effectiveness of TABI through extensive simulations. The whole approach can also be applied to other social media, with the purpose of maximizing overall participations, activities or attentions.

For future work, we plan to conduct user study to systematically verify our method, and to transcend the limitations of simulation. We will investigate heuristics that consider further influence cascades and find out the best timing for thread allocation. Furthermore, we will study the application of similar approaches to other social media that also possess rich interaction and social network data.

References

Anagnostopoulos, A.; Kumar, R.; and Mahdian, M. 2008. Influence and correlation in social networks. In *Proc. of KDD*.

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

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *Proc. of KDD*.

Dobzinski, S., and Schapira, M. 2006. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proc. of SODA*.

Goyal, A.; Bonchi, F.; and Lakshmanan, L. V. 2010. Learning influence probabilities in social networks. In *Proc. of WSDM*.

Gruhl, D.; Guha, R.; and et al. 2004. Information diffusion through blogspace. In *Proc. of WWW*.

Guo, L.; Tan, E.; and et al. 2009. Analyzing patterns of user content generation in online social networks. In *Proc. of KDD*.

Hogg, T., and Szabo, G. 2009. Diversity of user activity and content quality in online communities. In *Proc. of ICWSM*.

Horowitz, D., and Kamvar, S. D. 2010. The anatomy of a large-scale social search engine. In *Proc. of WWW*.

Ienco, D.; Bonchi, F.; and Castillo, C. 2010. The Meme Ranking Problem: Maximizing Microblogging Virality. In *SIASP 2010 workshop at ICDM 2010*.

Kempe, D.; Kleinberg, J.; and Èva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proc. of KDD*.

Kempe, D.; Kleinberg, J.; and Èva Tardos. 2005. Influential nodes in a diffusion model for social networks. In *Proc. of ICALP*.

Saito, K.; Kimura, M.; and et al. 2010. Selecting information diffusion models over social networks for behavioral analysis. In *Proc. of ECML PKDD*.

Saito, K.; Nakano, R.; and Kimura, M. 2008. Prediction of information diffusion probabilities for independent cascade model. In *Proc. of KES*.

Sarwar, B.; Karypis, G.; and et al. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW*.

Song, X.; Tseng, B. L.; and et al. 2006. Personalized recommendation driven by information flow. In *Proc. of SIGIR*.

Sun, T.; Chen, W.; and et al. 2010. Participation maximization based on social influence in online discussion forums. Technical Report MSR-TR-2010-142, Microsoft Research.

Tang, J.; Sun, J.; and et al. 2009. Social influence analysis in large-scale networks. In *Proc. of KDD*.

Vondrák, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. of STOC*.

Weng, J.; Lim, E.-P.; and et al. 2010. Twitterrank: finding topic-sensitive influential twitterers. In *Proc. of WSDM*.