Microscopic Social Influence

Ting Wang[†] Mudhakar Srivatsa[†] Dakshi Agrawal[†] Ling Liu[‡]
[†]IBM Research {tingwang, msrivats, agrawal}@us.ibm.com
[‡]Georgia Tech lingliu@cc.gatech.edu

Abstract

Social influences, the phenomena that one individual's actions can induce similar behaviors among his/her friends via their social ties, have been observed prevailingly in socially networked systems. While most existing work focuses on studying general, macro-level influence (e.g., diffusion); equally important is to understand social influence at *microscopic* scales (i.e., at the granularity of single individuals, actions, and time-stamps), which may benefit a range of applications. We propose μSI, a microscopic social-influence model wherein: individuals' actions are modeled as temporary interactions between social network (formed by individuals) and object network (formed by targets of actions); one individual's actions influence his/her friends in a dynamic, network-wise manner (i.e., dependent on both social and object networks).

We develop for μSI a suite of novel inference tools that enable to answer questions of the form: How may an occurred interaction trigger another? More importantly, when and where may a new interaction be observed? We carefully address the computational challenges for inferencing over such semantically rich models by dynamically identifying sub-domains of interest and varying the precision of solutions over different subdomains. We demonstrate the breadth and generality of μSI using two seemingly disparate applications. In the context of social tagging service, we show how it can help improve the accuracy and freshness of resource recommendation; in the context of mobile phone call service, we show how it can help improve the efficiency of paging operation.

1 INTRODUCTION

Cascading behavior, diffusion, and propagation of ideas, innovations, and information are fundamental processes taking place in socially networked systems [1, 2, 3]. It is well recognized that *social influence* is one complex and subtle force that governs these dynamics [4, 5]: the actions of one individual may induce his/her friends to behave in a similar way via their social connections. Interpreting an individual's behavior in his/her social context and correlating the actions of socially connected individuals are thus of tremendous interests from both analysis and design perspectives.

Recently social influence analysis has attracted intensive research interests, with examples including differentiating the effects of social correlation (e.g., homophily and confounding) and social influence on users'[1] activities [4], verifying the existence of correlation between personal behavior and social connection [5], finding the most influential nodes in social networks [6, 7], estimating the influence strength of social ties [8], and inferring influence channels for implicit networks [9]. Most of these studies focus on general, macro-level influence phenomena, irrespective of concrete users, actions, or time-stamps. We argue that equally important is to understand and model social influence at *microscopic* scales (i.e., individual users, actions, and time-stamps), which may carry significant benefits for a range of applications, such as

- *Information filtering.* With the advance of Web 2.0 technologies and social media, the amount of information received by normal users can easily go beyond their processing capacities (e.g., an average Twitter user receives over 93 tweets per day). Understanding at individual level how the browsing behaviors of users with similar interests affect each other may facilitate *personalized* recommendation to effectively filter uninteresting information and deliver high-quality information in time.
- *Mobile phone-call service.* Recent studies on human mobility [10, 11, 12] have revealed strong correlation between geographical and social distance of individuals (e.g., users tend to make more calls to closer friends within closer geographical distance). Modeling such influence at the level of individual users and locations offers valuable insight into characterizing user mobility and creating caller-callee profiles, which may significantly improve the efficiency of paging operation [13].
- *Targeted advertising.* Through "word of mouth" and product comparison, one user's purchase of a product may trigger his/her friends to buy functionally similar products, but with more personally favorable features [3, 14]. Clearly, accounting for such influence at the granularity of particular users and

[1]Following we use "user" and "individual" interchangeably.

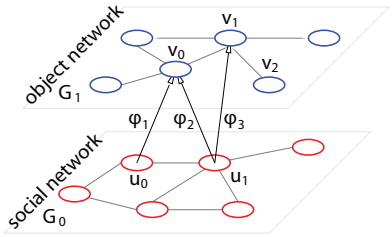

Figure 1: Example of social and object networks.

products is crucial for developing successful advertisement strategies.

Not surprisingly, modeling microscopic social influence is in general a difficult task, featuring a set of unique challenges: (i) the model should account for the distinct characteristics of individual users, actions, and time-stamps, as well as their complex interconnections; (ii) it should support application-specific, micro-level influence mechanisms (e.g., how one action induces another concretely); (iii) it should be accompanied by scalable inference tools to track the influence of occurred actions and predict future actions. None of these challenges are trivial. Consider challenge (iii) for example: given that we target the level of individual users and actions, we need to consider all the potential actions by all the users when performing inference, which implies prohibitive computational complexity for large user or action space. This work, to our best knowledge, represents the first attempt to model microscopic social influence at the granularity of individual users, actions, and time-stamps.

For the first challenge, in contrast of alternative models such as bipartite network or heterogenous network [15], we define a more general model (shown in Figure 1) wherein: it replicates all individual users and objects (targets of actions, e.g., weblogs), whose interconnections respectively form social network and object network; meanwhile, actions are modeled as temporary interactions between these two networks, which reflect their dynamic nature.

For the second challenge, we propose a novel *heat field over product network* (HFPN) model to encode the concrete influence mechanism. Intuitively in the "product" of social and object networks, each node represents one potential action, while each edge represents the "influence channel" between two actions. The influence is naturally modeled as "heat" that flows through such channels. The flexibility of this model lies in its potentially unlimited number of ways to specify the influence channels and strengths.

For the third challenge, we develop a complete library of inference operations that capacitate to continuously *track* the state of "heat distribution" over HFPN model, *update* the state once new actions are observed, and *predict* where and when new actions may occur. We carefully address the computational challenges for inferencing over HFPN model by drawing analogy to finite element analysis (FEA), where the key idea is to vary the precision of numerical methods over a physical domain. In this paper, we present solutions that dynamically identify sub-domains of interest (over the entire space of potential actions), and vary the precision of influence estimation around interesting sub-domains.

We entitle this complete framework as μSI. We conduct an empirical evaluation in two seemingly disparate applications using real-life datasets. In the case of social tagging service (e.g., Del.icio.us), we perform web resource recommendation based on the prediction of μSI, which demonstrates significant improvement over conventional solutions in terms of accuracy and freshness of recommendation. In the case of mobile phone call service, we apply μSI to improve the efficiency of paging operation by reducing the number of cells to search. It is shown that μSI enhances the state-of-the-art callee profile-based approach [13] by reducing about 25% of signaling traffic.

The remainder of the paper is organized as follows. Section 2 surveys relevant literature; Section 3 introduces the fundamental concepts and building blocks of μSI; a library of inference operations are presented in Section 4, in which we also detail their scalable implementation; we present the empirical evaluation of μSI in Section 5; the limitations of μSI and future research directions are discussed in Section 6.

2 RELATED WORK

Network dynamics has been a long-lasting topic for network science. Existing research mostly focuses on studying network dynamics within the setting of a single, homogenous network, with examples including diffusion and propagation [1], influence maximization [2, 7], community formation [16], and network evolution [17, 18]. Recently intensive research efforts have been directed to social influence in online networks [4, 8]. Using the terms of our microscopic influence model, these works essentially focus on the interactions between one (implicit) object and multiple users.

More recently, a few works have started to take account of the rich information conveyed by the interactions between multiple objects and users, to detect communities [19], to analyze the evolution of both object and social networks [20], or to model topic-sensitive social influence [21]. In all these cases, however, the relationships between users and objects are modeled as a heterogenous network wherein user-object interactions are treated as static links. This formation, even though amenable to graphical model-based learning, ignores the dynamic nature of interactions (i.e., time-sensitivity), which we deem as one key element in understanding and modeling microscopic social influence.

The work closest to ours is perhaps [22], in which the authors attempt to predict the social influence of a specific user-object interaction, i.e., how many other

users will interact with the specific object. We target a more general setting wherein multiple interactions may exert their influence simultaneously. Moreover the time sensitivity of social influence is ignored in [22].

There are several other lines of work we build upon. Heat field model on a manifold has been applied for classification [23] and spam-resilient ranking [24]. Meanwhile, product network, especially Kronecker product has been applied to model graph generation [25] and dynamic tensor analysis [26]. To the best of our knowledge, however, the present work is the first that considers the optimization of heat field model in the context of product network.

3 FUNDAMENTALS

This section introduces the basic building blocks of μSI.

3.1 Preliminaries

We start with introducing a set of fundamental concepts used throughout the paper. We consider two types of entities, "users" (subjects of actions, e.g., Internet users) and "objects" (targets of actions, e.g., weblogs). We first formalize the concept of *interaction*.

Definition 1 (Interaction/Action)**.** *An interaction (or action) is a temporary (say, occur at a specific timestamp t) association between a user u and an object v, denoted by $\phi(u, v, t)$.*

Examples include Internet users reading weblogs and mobiles connecting to base stations. We consider such interactions within the context of *networks*.

Definition 2 (Network)**.** *A network is modeled as a (directed) graph $G = (V, E)$ where V and E represent a set of entities and their interconnections, respectively. Each $e \in E$ may be further associated with weight $w(e)$ specifying the strength of such interconnection.*

The interconnections in social and object networks may convey different meanings. In social network, an interconnection between two users indicates their social tie, which functions as the channel of social influence; while in object network, an interconnection between two objects indicates their proximity in certain sense, e.g., semantic similarity of two weblogs (as reflected in their references to each other), or geographical closeness of two cellular towers. Interestingly, such proximity may also become the channel via which one interaction influences another. For example, after reading one weblog, a user may further read another relevant one by following their reference.

3.2 Building Blocks

To model the influence of occurred interactions $\{\phi\}$ on triggering (potential) interactions $\{\phi'\}$, we essentially need to capture (i) how the influence of $\{\phi\}$ is passed to $\{\phi'\}$, (ii) how much influence is transferred, and (iii) how the accumulated influence at $\{\phi'\}$ triggers their occurrence.

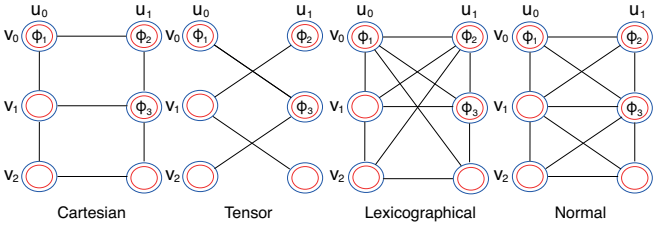

Figure 2: Examples of product networks.

3.2.1 Product Network Model

The structures of both social and object networks constrain the locality of the influence between interactions, i.e., interaction $\phi = (u, v, t)$ tends to affect "neighboring" users of u or objects of v.

Example 1. In Figure 1, assuming G_0 as an online social network and G_1 as a blogosphere, after reading weblog v_0, user u_0 may further read v_1 if interested in the topic covered by v_0. Meanwhile, knowing that his/her friend u_0 has read v_0, user u_1 may also be interested in reading it. Therefore the interaction (u_0, v_0) triggers both interactions (u_0, v_1) and (u_1, v_0).

It is clear that such concrete "influence channels" are highly application-specific: the influence may spread over a single network or over both networks simultaneously, or may feature different spread rates within two networks. Instead of attempting for a one-size-fit-all model, we enable a class of influence channel instantiations based on the concept of product network.

Definition 3 (Product Network)**.** *For two networks $G_0 = (V_0, E_0)$ and $G_1 = (V_1, E_1)$, their product network is defined as a graph $G_\times = (V_\times, E_\times)$ such that node $\phi = (u, v) \in V_\times$ if $u \in V_0, v \in V_1$, while the existence of edge $\phi_i - \phi_j$ $((u_i, v_i) - (u_j, v_j))$ can be specified by* **any** *logical statements of $u_i - u_j$, $v_i - v_j$, $u_i = u_j$, and $v_i = v_j$.*

As an example, Tensor product ($G_\otimes$) is defined as $\phi_i - \phi_j$ $((u_i, v_i) - (u_j, v_j))$: $u_i - u_j \in E_0$ and $v_i - v_j \in E_1$.

In the product of two networks G_0 and G_1, each node $\phi = (u, v)$ essentially corresponds to a (potential) interaction between $u \in G_0$ and $v \in G_1$, while each edge specifies an influence channel between two interactions. The flexibility of the specification of edges allows application-dependent instantiation, as shown in the next example.

Example 2. Figure 2 shows a subset of possible product networks corresponding to two sub-networks $\{u_0, u_1\} \subset G_0$ and $\{v_0, v_1, v_2\} \subset G_1$ in Figure 1.

A critical question is how to choose the "right" product model for the intended application. To answer this, we need to understand the influence propagation pattern by observing historical data (details in Section 5). In Example 1, we may choose Cartesian product because it is observed that the influence propagates between connected users (with respect to a specific weblog) or between connected weblogs (with respect to a

specific user), but not simultaneously. Another question is how to determine the weight w_{ij} of edge $\phi_i - \phi_j \in E_\times$, which essentially specifies the influence spread rate from ϕ_i to ϕ_j. In this paper we set w_{ij} based on the weight w^0_{ij} and w^1_{ij} of corresponding edges $u_i - u_j$ and $v_i - v_j$ in G_0 and G_1. We will detail the setting of parameters in Section 5.

3.2.2 Heat Field with Update Model

Heat diffusion [23] is a physical phenomenon wherein heat flows from positions with high temperature to that with low temperature. It has been observed [24, 27] that this is a natural way to describe the influence spreading from occurred interactions to other (potential) interactions: occurred interactions can be considered as heat sources with high temperature, and their influence is modeled as heat that flows to other (potential) interactions with low temperature through the underlying geometric structure of product network; the accumulated heat at a potential interaction indicate its tendency to happen; the initial heat sources (occurred interactions) and geometric structures (influence channels) determine the heat distribution.

More specifically, let $f_i(t)$ represent the heat amount (or temperature) at node[2] ϕ_i at time t. During a tiny time period $[t, t + \Delta t]$, we consider two types of heat movement.

- The heat node ϕ_i receives from or transfers to its neighbor ϕ_j through edge (influence channel) $\phi_i - \phi_j$, $T(i, j, t, \Delta t)$ over period Δt. Intuitively, $T(i, j, t, \Delta t)$ should be proportional to the difference $(f_j(t) - f_i(t))$, the elapsed time Δt, and the weight w_{ij} of edge $\phi_i - \phi_j$ (which specifies the influence spread rate over $\phi_i - \phi_j$).
- The heat of ϕ_i diffuses to the environment surrounding the network, $D(i, t, \Delta t)$, which captures the intuition that the influence decays with time, even without being transferred to other interactions. We assume that $D(i, t, \Delta t)$ is proportional to the heat $f_i(t)$ and elapsed time Δt.

Combing the two components above, we can formulate the heat change at node ϕ_i at time t as follows:

$$\begin{aligned} & f_i(t + \Delta t) - f_i(t) \\ = \; & \textstyle\sum_{\phi_i - \phi_j} \alpha w_{ij}(f_j(t) - f_i(t))\Delta t - \beta f_i(t)\Delta t \end{aligned}$$

where α and β denote respectively the diffusion rate inside and outside the network. Note that here we consider the product network as an undirected graph, while the discussion can be readily generalized to the case of directed or probabilistic network. Expressed in a matrix form:

$$\mathbf{f}(t + \Delta t) - \mathbf{f}(t) = (\alpha(W - D) - \beta I)\mathbf{f}(t)\Delta t$$

[2]In the context of product network, we consider the terms "node" and "interaction" interchangeable.

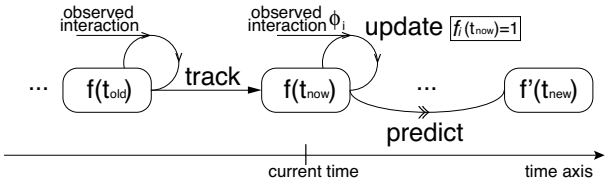

Figure 3: Operations of μSI.

where W denotes the weight matrix of product network, the diagonal matrix D is defined as $D_{ii} = \sum_{\phi_i - \phi_j} W_{ij}$, and I an identify matrix. Let $H = W - D$. With $\Delta t \to 0$, we have the solution: $\mathbf{f}(t) = e^{t(\alpha H - \beta I)}\mathbf{f}(0)$. We name the heat distribution over the network as the *heat field* (HF).

It is noticed that this is a *time-invariant* system, i.e., for $\mathbf{f}(t_1) = \mathbf{f}(t_2)$, we have $\mathbf{f}(t_1 + \Delta t) = \mathbf{f}(t_2 + \Delta t)$. Based on this nice property, we can readily incorporate the mechanism of *update* in our model: an update at t replaces $\mathbf{f}(\cdot)$ with a (possibly arbitrary) configuration $\mathbf{f}^*$, i.e., starting from t, the system evolves with $\mathbf{f}^*$ as the initial condition, that is

$$\mathbf{f}(t + \Delta t) = e^{\Delta t(\alpha H - \beta I)}\mathbf{f}(t) \to e^{\Delta t(\alpha H - \beta I)}\mathbf{f}^* \tag{3.1}$$

We entitle the combination of product network and heat diffusion with update models as *heat field over product network* (HFPN). Using HFPN, we can continuously track the heat distribution over the product network; for newly occurred interactions, we consider them as new heat sources, and update the heat field accordingly. In general the initial heat should be set according to the *influence capacity* of corresponding users or objects (e.g., users who have more friends tend to have more capacity). In this paper, without loss of generality, we assume that each new heat source is initiated with one unit of heat.

3.2.3 Invocation Model

The accumulated influence at node ϕ_i at given time t indicates the tendency that the corresponding interaction occurs at t. Yet, we need an invocation model that describes how the accumulated influence at ϕ_i actually triggers its occurrence.

We assume that with heat as $f_i(t)$, ϕ_i is activated with probability $g(f_i(t) - \lambda)$, where λ is a threshold. The function $g(\cdot)$ has the property that $g(x) = 0$ if $x \leq 0$, and is non-decreasing as x increases.

The concrete formation of invocation model should be empirically validated and parameterized using historical data in intended application, with details discussed in Section 5.

4 TRACKING, UPDATING, PREDICTING

Atop the heat field over product network and invocation models, we now construct the complete framework of μSI, which enables to track, update, and predict interactions.

4.1 A Complete Framework

Starting with the heat field of product network $G_\times$ of G_0 and G_1 at any given time, μSI capacitates us to perform the following three operations:

- Track. Given the estimation of heat field $\mathbf{f}(t_{old})$ at time t_{old} when the latest interaction is observed, we estimate the current heat field $\mathbf{f}(t_{now})$;
- Update. For an interaction ϕ that is observed at current time t_{now}, we update the estimation of heat field $\mathbf{f}(t_{now})$ accordingly;
- Predict. Based on the estimation regarding current heat field $\mathbf{f}(t_{now})$, we predict the heat field state $\hat{\mathbf{f}}(t_{new})$[3] at a future time t_{new}.

The relationships among these operations are illustrated in Figure 3. Equipped with the three operations, we are able to query the state of heat field at arbitrary time-stamp[4].

4.2 Implementation

The concepts of *Track* and *Update* in the HFPN model are fairly straightforward. Specifically, for tracking, given previous state $\mathbf{f}(t_{old})$, the current heat field can be estimated as $\mathbf{f}(t_{now}) = e^{(t_{now}-t_{old})(\alpha H-\beta I)}\mathbf{f}(t_{old})$; for updating, given an observed interaction $\phi_i = (u_i, v_i, t_{now})$, we consider it as a new heat source, and update $f_i(t_{now}) = 1$.

Directly evaluating this model features prohibitive computational complexity due to the costly matrix exponential operation. In general, for matrix exponential e^M, we may resort to a discrete approximation: $e^M \approx (I + M/N)^N$, where N is the number of iterations that controls the error $||(I+M/N)^N - e^M||$. Next we expose possible solutions that further simplify the computation by (i) fully exploiting the structural properties of product network and (ii) intelligently caching and reusing invariant intermediate results. Due to space constraint, we use Cartesian product as an example, while our discussion can be extended to other types of product networks.

4.2.1 Decomposition of Product Network

Note that Eq. (3.1) can be re-written as

$$\mathbf{f}(t + \Delta t) = e^{-\Delta t(\alpha D+\beta I)}e^{\Delta t\alpha W}\mathbf{f}(t)$$

While $e^{-\Delta t(\alpha D+\beta I)}$ ($(\alpha D + \beta I)$ is diagonal) is easy to compute, we focus on $e^{\Delta t\alpha W}\mathbf{f}(t)$. In the case of Cartesian product, the weight matrix W is defined as a Kronecker sum $W = W_0 \otimes I + I \otimes W_1$, where W_i $(i = 0, 1)$ represent the weight matrices of component networks. The matrix exponential of Kronecker sum has a nice property.

[3]We use $\hat{f}$ and f to distinguish the estimates using predicting and tracking, respectively.

[4]Note that essentially we can also derive a "Backtrack" operation that takes current state $\mathbf{f}(t_{now})$ as input, and estimates the heat field Δt ago: $\bar{\mathbf{f}}(t_{now} - \Delta t)$.

Property 1. *For matrices M_0 and M_1, the exponential of their Kronecker sum satisfies $e^{M_0 \oplus M_1} = e^{M_0} \otimes e^{M_1}$.*

We therefore have the following transformation:

$$e^{\Delta t\alpha W} = e^{\Delta t\alpha W_0} \otimes e^{\Delta t\alpha W_1}$$

Let $\mathbf{m} = \mathsf{vec}(M)$ denote the *vectorization* of matrix M formed by stacking the columns of M into a single column vector $\mathbf{m}$, and $M = \mathsf{dvec}(\mathbf{m})$ be the inverse operation. We have the next property.

Property 2. *For three (multiplication compatible) matrices M_0, M_1, and X, the following vectorization property holds: $(M_0 \otimes M_1)\mathsf{vec}(X) = \mathsf{vec}(M_1 X M_0^T)$.*

Clearly, the matrix-vector multiplication can be computed in $O(n^3)$ time for $|M_i| = n^2$ $(i = 0, 1)$. Especially, when M_0 and M_1 are sparse (which is typically the case), it can be more efficient: if there are $O(n)$ non-significant entries in M_0 and M_1, the computation takes only $O(n^2)$ time. Thus for Cartesian product network, we have the following transformation of heat field equation:

$$\mathbf{f}(t + \Delta t) = e^{-\Delta t(\alpha D+\beta I)}\mathsf{vec}(e^{\Delta t\alpha W_1}\mathsf{dvec}(\mathbf{f}(t))e^{\Delta t\alpha W_0^T})$$

4.2.2 Caching of Intermediate Results

It is noticed that in the heat field equation for given $\mathbf{f}(t)$, α, and β, $\mathbf{f}(t + \Delta t)$ is essentially a function of Δt.

Clearly, if the time dimension can be discretized into time-ticks δt, we can essentially cache the intermediate results $e^{\Delta t(\alpha H-\beta I)}$ for $\Delta t = \delta t, 2\delta t, \ldots$, and reuse them in state estimation. In following we use $\kappa(i)$ to denote the "kernel" $e^{i\delta t(\alpha H-\beta I)}$. The saving of computation, however, comes at the expense of storage. We introduce two optimization strategies to strike a balance between computation and storage costs.

Instead of storing $\kappa(i)$ for all i's from 1 to m, we may selectively cache a subset of i's in a *geometric* manner, i.e., $i = 2^0, 2^1, \ldots, 2^{\log_2 m}$ (assuming m is an exponential of two). We can now compute $\mathbf{f}(t+k\delta t)$ for any $k \in [1, m]$ as the multiplication of less than $\log_2 m$ matrices $\kappa(\cdot)$ and vector $\mathbf{f}(t)$.

Next we introduce the second optimization strategy, partial materialization. Instead of caching entire matrix exponential, we may only need to cache its core part, which, in conjunction of geometric increment, leads to a both space and computation efficient scheme. Recall that in Cartesian product, the heat field function is defined as $\mathbf{f}(t+k\delta t) = e^{-k\delta t(\alpha D+\beta I)}e^{k\delta t\alpha W_0} \otimes e^{k\delta t\alpha W_1}\mathbf{f}(t)$. Clearly, the term $e^{-k\delta t(\alpha D+\beta I)}$ is directly computable; while the part $e^{k\delta t\alpha W_0} \otimes e^{k\delta t\alpha W_1}$ can be re-written as:

$$\begin{aligned} e^{k\delta t\alpha W_0} \otimes e^{k\delta t\alpha W_1} &= (\prod_{i=1}^{j} e^{2^{k_i}\delta t\alpha W_0}) \otimes (\prod_{i=1}^{j} e^{2^{k_i}\delta t\alpha W_1}) \\ &= (\prod_{i=1}^{j} \kappa_0(k_i)) \otimes (\prod_{i=1}^{j} \kappa_1(k_i)) \end{aligned}$$

where we only cache the part $\{\kappa_0(x)\}$ and $\{\kappa_1(x)\}$ $(x = 0, \ldots, \log_2 m - 1)$, reducing the storage cost to $O(n^2 \log m)$. Meanwhile, the computation now consists of no more than $(2\log_2 m + 2)$ matrix multiplications, thus featuring complexity of $O(n^{2.376} \log m)$ if the state-of-the-art matrix multiplication algorithms are applied.

4.3 More on Prediction

The *Predict* operation essentially takes as input the current state $\mathbf{f}(t_{now})$, and attempts to estimate a future state $\hat{\mathbf{f}}(t_{new})$. The difficulty lies in taking account of those interactions that may occur during the time interval $(t_{now}, t_{new}]$. These potential interactions (invocations) may make the actual result significantly deviate from the invocation-agnostic prediction $e^{(t_{new}-t_{now})(\alpha H-\beta I)}\mathbf{f}(t_{now})$.

A naïve solution would be a Monte Carlo simulation-based scheme: we divide the time axis into a sequence of high-resolution "time-ticks", each of length δ_t: at the $(i+1)$-th tick, we estimate $\mathbf{f}(t+(i+1)\delta_t)$ based on the simulation result of $\mathbf{f}(t+i\delta_t)$ (essentially a tracking operation); for an interaction ϕ_i with heat above λ, we activate it with probability according to $g(f_i(t+(i+1)\delta_t)-\lambda)$ in the invocation model (essentially a sampling operation). While intuitive and faithful to the invocation mechanism, this scheme suffers scalability issues especially when the network scale is large or the granularity of time-tick is small, because it involves the tracking and sampling operations at each time-tick and possibly for all potential interactions. We consider this scheme as the baseline solution in comparison.

We attempt to construct a scheme that provides flexible trade-off between the cost of tracking and sampling operations and the quality of simulation results. We start with defining the concept of *activation window*.

Definition 4 (Activation Window)**.** *For given threshold λ and interaction ϕ_i, a time interval $[t_s, t_e]$ is called an activation window of ϕ_i if (i) $f_i(t) > \lambda$ for $t \in (t_s, t_e)$ and (ii) $\nexists[t'_s, t'_e] \supset [t_s, t_e]$ has such property. The peak value $\max_t f_i(t)$ $(t \in [t_s, t_e])$ is called the window height, and t_s and t_e the starting and ending bounds of the window.*

According to the invocation model, inside each activation window, the potential interaction obtains the opportunity to be activated. A model faithful to the exact simulation guarantees that such opportunities are fully respected. We therefore strive for an approximate objective: *each interaction is given (with high probability) one activation opportunity during each of its windows; while the probability is positively correlated with the window height.* This is similar to Finite Element Analysis (FEA) in spirit.

Based on this approximation, we propose a "lazy-probing" scheme that allows to skip state estimation during a "safe window" without denying the invocation opportunities for highly likely interactions. To do so, we first need to understand the maximum length of such safe window. We have the following theorem:

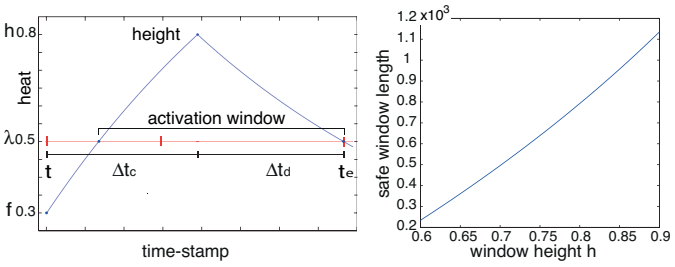

Figure 4: Charge and discharge phases of an activation window, and length of safe window with respect to window height h (default setting: $\alpha = 0.2$, $\beta = 0.05$, $\lambda = 0.5$, $f = 0.3$, $h = 0.8$, $d = 2$).

Theorem 1. *At time t, for an interaction ϕ_i with current heat f $(f < \lambda)$, degree d, the ending bound t_e of its next activation window with height above h must satisfy:*

$$t_e - t \geq \frac{1}{\alpha d + \beta} \ln \frac{\alpha d - (\alpha d + \beta) f}{(\alpha d - (\alpha d + \beta) h)(\lambda + 1 - h)}$$

Proof (Theorem 1). For simplicity of exposition, we assume that each occurred interaction is initiated with one unit of heat, and the weights of all the edges are uniformly set as one. Assume that the next appearing activation window has height h. We divide the time interval $[t, t_e]$ into two phases, *charge* and *discharge*. Let Δt_c and Δt_d be their length. In the charge phase, the heat of ϕ increases; after reaching the expected height h, it enters the discharge phase, and the heat drops below λ. These concepts are illustrated in Figure 4. We intend to find the lower bounds of Δt_c and Δt_d. Clearly, $t_e - t \geq \Delta t_c + \Delta t_d$.

First notice that the lower bound of Δt_c is achieved when all of neighboring interactions have maximum possible heat "1". In this case, the heat equation for the charge phase can be written as $\frac{\mathrm{d}f_i}{\mathrm{d}t} = \alpha d(1 - f_i) - \beta f_i$, with initial condition $f_i(0) = f$ and end condition $f_i(\Delta t_c) = h$. Solving the differential equation, we get $\Delta t_c \geq \frac{1}{\alpha d+\beta} \ln \frac{\alpha d-(\alpha d+\beta)f}{\alpha d-(\alpha d+\beta)h}$. Meanwhile, the lower bound of Δt_d is achieved when ϕ_i receives no heat, but diffuses its heat to neighboring nodes and the media. We have the corresponding heat equation as $\frac{\mathrm{d}f_i}{\mathrm{d}t} = -(\alpha + \beta) f_i$, with initial condition $f_i(0) = h$ and end condition $f_i(\Delta t_d) = \lambda$. Solving the equation gives us: $\Delta t_d \geq \frac{1}{\alpha d+\beta} \ln \frac{1}{\lambda+1-h}$. Combining these two inequalities leads to the result. The overall behavior of $(\Delta t_c + \Delta_d)$ with respect to window height h is illustrated in the right plot of Figure 4, which shows approximately linear correlation.

Atop Theorem 1, we introduce our lazy-probing prediction scheme (Algorithm 1). At each examined time-tick t', we attempt to activate each potential interaction

Algorithm 1: Lazy-probing prediction.

Input: current time t_{now}, current state $\mathbf{f}(t_{now})$, future time t_{new}, activation threshold λ, activation window height h
Output: estimated future state $\hat{\mathbf{f}}(t_{new})$

$t' \leftarrow t_{now}$, $\hat{\mathbf{f}}(t') \leftarrow \mathbf{f}(t_{now})$;
while $t' < t_{new}$ **do**
  $\Delta t \leftarrow t_{new} - t'$;
  foreach $\phi_i \in V_\times$ **do**
    if $\hat{f}_i(t') \geq \lambda$ **then**
      `// sampling (activation trial)`
      if *i is not activated* **then**
        set $\hat{f}_i(t') = 1$ with prob. $g(\hat{f}_i(t') - \lambda)$;
    else
      `// estimate minimum skippable interval`
      $\Delta t \leftarrow \min\{\Delta t, \frac{1}{\alpha d+\beta} \ln \frac{\alpha d-(\alpha d+\beta)\hat{f}_i(t')}{[\alpha d-(\alpha d+\beta)h](\lambda+1-h)}\}$
  `// update estimation`
  $\hat{\mathbf{f}}(t' + \Delta t) \leftarrow e^{\Delta t(\alpha H-\beta I)}\hat{\mathbf{f}}(t')$;
  $t' \leftarrow t' + \Delta t$;

ϕ_i with heat above threshold λ according to probability $g(\hat{f}_i(t') - \lambda)$; for potential interactions with heat below λ, we estimate the maximum skippable interval according to Theorem 1; the minimum of all such skippable intervals is considered as the global skippable interval Δt, and the state estimation is updated for time $(t' + \Delta t)$. This process repeats until $t' = t_{new}$.

5 APPLICATION & EVALUATION

Following we empirically evaluate the efficacy of μSI in two concrete applications, namely, social tagging service and mobile phone call service, using real-life datasets.

The first dataset is the social network corresponding to a set of IBM employees who participated in the SmallBlue project [28]. The dataset comprises two snapshots of the network as of January 2009 and July 2009, involving 41,702 and 43,041 individuals, respectively. Without further information, all links of the network are set with uniform weight 1.0. It also contains an archive of bookmarks tagged by the individuals appearing in the SmallBlue project, collected by Dogear[5], a personal bookmark management application. The archive contains 20,870 bookmark records, with respect to 7,819 urls. Since the urls are fully anonymized, we solely use their associated tags define their semantic relationships and to construct the url network. Let $\mathit{tag}(r)$ denote the set of tags associated with url r. Specifically, we define the weight w_{ij} of the link $r_i r_j$ using their *Jaccard similarity coefficient*:

$$w_{ij} = w_{ji} = |\mathit{tag}(r_i) \cap \mathit{tag}(r_j)|/|\mathit{tag}(r_i) \cup \mathit{tag}(r_j)|$$

The second dataset was collected from the Reality Mining dataset [29] project. It consists of the communication logs of 94 users over the period from September 2004 to June 2005, with information including location logs (nearest base stations), phone call logs, social relationships (friends or non-friends). We use the cell transition information in location logs to infer the cell network (totally 3,138 cells), and use the friend and non-friend relationships to construct the social network. Without further information, the links in both networks are of uniform weight 1.0. We extract the part of logs corresponding to communications (voice, data, SMS) between the 94 users. The resulting archive contains 3,984 communication records.

In this paper we set the edge weight w_{ij} in the product network based on the weight w^0_{ij} and w^1_{ij} of corresponding edges $u_i - u_j$ and $v_i - v_j$ (or nodes) in G_0 and G_1. We use Tensor and Normal product networks in our experiments (the selection of product network type will be discussed in concrete case studies). For Tensor product network $G_\otimes$, a natural way of doing so is to set $w_{ij} = w^0_{ij} w^1_{ij}$; intuitively, we assume that the influence spread rate between two neighboring interactions $\phi_i = (u_i, v_i)$ and $\phi_j = (u_j, v_j)$ is proportional to the similarity of u_i and u_j, and that between v_i and v_j. While for Normal product network which allow "stationary walk" in either one of the networks (e.g., u_i and u_j or v_i and v_j can be equivalent), we need to determine the weight of self-loop, e.g., $u_i - u_i$. If no further information is available, we use a non-informative setting 1.0.

We implement our models in Matlab and conduct all the experiments on a workstation running Linux with 4G RAM and 2GHz Intel Core 2 Duo.

5.1 Case Study 1: Resource Recommendation in Social Tagging Service

Social tagging services (e.g., Del.icio.us, CiteULike, etc.) allow everyday users to annotate online resources (e.g., urls) with freely chosen keywords (tags), which provide meaningful collaborative semantic data that can be exploited by recommender systems. However, performing resource recommendation based on tagging data is a challenging problem. First, it involves multi-type, complexly interrelated entities, namely, users, tags, and resources; the available tagging data in the form of user-tag-resource may only expose a fairly sparse set of views of the hidden semantic correlation between users and resources. Second, the tagging behavior of users may demonstrate strong time sensitivity, e.g., most tags of a resource may be generated during a short time period when the resource features an ongoing hot topic among users; while the historical tagging records of a user may not reflect his/her current interest. Most existing solutions fail to address these two aspects [30, 31, 32].

[5] http://www.ibm.com/dogear

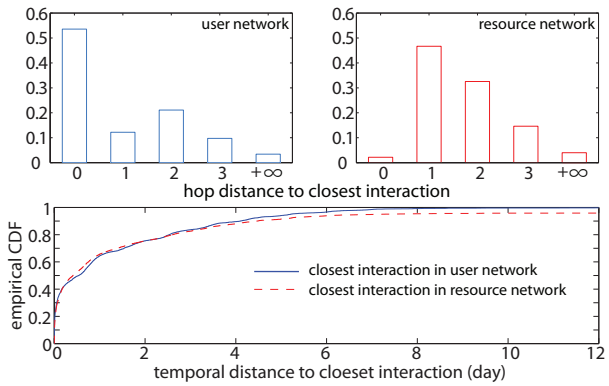

Figure 5: Spatial (in user and resource networks) and temporal locality of influence of tagging actions.

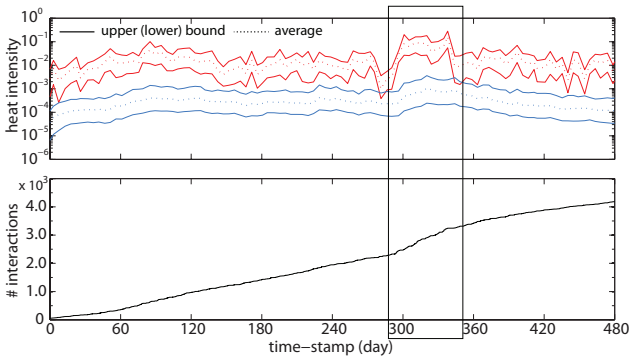

Figure 6: Estimated heat intensity of observed interactions and inactive (background) interactions.

With the help of the microscopic social influence model, we provide a *dynamic, social* perspective of online resource recommendation. We consider a tagging action as a dynamic interaction between the social and resource networks. We have the following assumptions. First, the tagging action of a specific user regarding a specific resource indicates his/her ongoing interest in the resource. Second, the influence of a tagging action may spread over both social and resource networks: (i) socially close users tend to share common interests, and may follow peers to tag the same resource (which is frequently observed in social networking sites); (ii) meanwhile, users tend to tag similar resources if they find the topic covered by these resources interesting.

Based on these two observations, we propose a novel resource recommendation approach for social tagging service. Intuitively, we use the social influence model to capture the effect of specific tagging actions on inducing other tagging actions. Leveraging this model, we may predict the tagging actions of potential interested users before the actions actually occur, such that the resources can be recommended to these users in advance of time.

Assumption Validation and Model Instantiation We start with analyzing the dataset to validate the assumptions of our solution. Specifically three important assumptions are made in our model: (i) the influence of occurred interactions on triggering other interactions is local to both networks; (ii) the influence may exhibit different spread characteristics in the two networks (captured by the concrete instantiation of production network and parameterization); and (iii) such influence tends to decay over time.

We examine the entire history of tagging actions that occurred during the time period from April 20, 2008 to August 20, 2009. For each interaction (u, v) (u in user space and v in resource space), we search for its closest neighboring interaction (u', v') in the history before (u, v) along the dimensions of user and resource (within a reasonable time window, say 30 days), i.e., $\min_{(u',v')} \texttt{dist}(u, u') \vee \texttt{dist}(v, v')$, and measure this closest (hop) distance.

The upper plot of Figure 5 shows the distribution of such shortest distance in both user and resource networks, where hop-0 refers to the node itself (e.g., one user tags several relevant resources or one blog is tagged by several relevant users), and hop-∞ includes all the cases of shortest distance above 4 hops. It is noticed that both distributions demonstrate strong locality: a majority of closest neighbors belong to the hop-0 or hop-1 categories, and the distribution decays as hop increases. Further, the distributions exhibit significantly different characteristics in the two networks: the hop-0 category dominates in the social network, while the hop-1 category accounts for the majority in the resource space. This may be explained by that the influence in resource space is more "dynamic" than that in user space, i.e., the tendency that a specific user looks for resources with similar topics is stronger than that he/she is influenced by the interests of other users. Leveraging this observation, in our implementation, we adopt a Normal product network which intuitively captures hop-0 influence in both networks (see Figure 2), but assign the resource network with larger diffusion rate. In particular, after testing three sets of setting of diffusion rates α in both networks: 0.1, 1, and 10, we fix the diffusion rate $\alpha = 0.1$ and 10 respectively for the user and resource networks for its best performance. By default, the time-tick is fixed as 10^8 ms (approximately a day).

One questions remains: what are the temporal characteristics of such closest neighboring interactions? To answer this, we measure the temporal distance between the interactions (u, v) and (u', v'), and examine the distribution of the closest interactions along the user and resource dimensions, with results shown in the lower plot of Figure 5. It is clear that (i) both distributions match fairly well, indicating that with high chance the closest neighboring interactions in the two networks are essentially the same one, and (ii) both distributions decay quickly as the temporal distance grows, implying the strong temporal locality of influence.

Predictive Power
This set of experiments evaluate the predictive power of

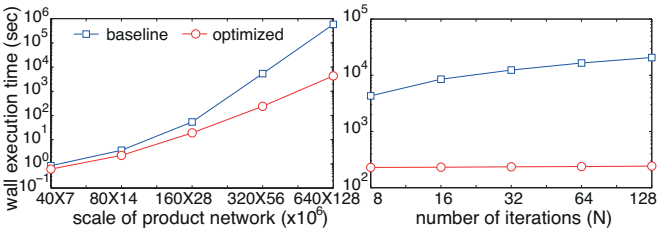

Figure 7: Running time of Track operation with respect to varying network scale and precision setting.

our social influence model, i.e., its capability of differentiating interactions highly likely to occur from the rest "inactive" ones. We run the tracking process. Specifically at the t-th time-tick, we update the estimation $\mathbf{f}(t)$ of the heat field based on the observed interactions at t, based on which we predict $\mathbf{f}(t+1)$ for the $(t+1)$-th tick. We then compare the estimated heat intensity of those interactions actually observed at $(t+1)$ and the rest inactive ones.

The results (mean and deviation) are shown in the upper plot of Figure 6 (in logarithmic scale[6]). It is observed that the two bands are clearly separable, indicating the predictive power of our model: the heat of actually observed interactions is typically two orders of magnitude higher than that of inactive ones. It is also observed that there exists certain "fluctuation" at the heat estimates (for both observed and inactive interactions) for the approximate time period of 290th $\sim$ 350th day, as highlighted in box. We then examine the overall growth of number of interactions, as shown in the lower plot of Figure 6. It is seen that for the corresponding period, there exists a burst of tagging actions. We therefore conclude that (i) the absolute heat intensity measure is sensitive to the occurrence rate of interactions, which tends to evolve over time; yet (ii) the relative heat intensity (over the average level) may still serve as a good indication of the occurrence likelihood of interactions.

Scalability

This set of experiments are designed to evaluate the operation efficiency of μSI, by measuring the execution time of *Track*, the most fundamental operation of μSI, with respect to varying network scale and precision requirement. More specifically, we apply the *KronFit* tool in Stanford Network Analysis Platform[7] (SNAP) to extract the "cores structures" of user and resource networks, and apply the *KronGen* tool in SNAP over such cores to generate networks of different scales; also we generate different precision settings by varying the number of iterations required in computing the discrete approximation of matrix exponential. We evaluate both baseline and optimized versions of Track operation, with results shown in Figure 7. As predicted by the theoretical analysis, the optimized Track operation achieves orders of magnitude of performance gain over the baseline implementation, and can easily scale up to product network of scale 8.2×10^{11} or hundreds of iterations.

Prediction Accuracy versus Execution Efficiency

Following the estimation regarding the state of heat field, the next step is to accurately predict the interactions that are highly likely to occur. The invocation model bridges the gap between the two. The concrete form of the invocation model is typically application-dependent; in this set of experiments, we consider one possible construction for the application of resource recommendation.

At each time-tick t, we measure the estimated heat intensity for both observed interactions and inactive ones. We plot the results in Figure 8. The left plot shows the distribution of the heat intensity of both occurred and inactive interactions. Note that the measures here have been normalized to the average heat intensity, i.e., the average level as 0 (in logarithmic scale). One can notice the significant statistical difference between observed and inactive interactions. For each specific heat level (in logarithmic scale), we then measure the relative frequency of observed interactions over the total number of interactions at that level. The right plot of Figure 8 shows the result (in logarithmic scale). We use an exponential curve to fit the relative frequency, which shows a tight match with the data. We use this fitted model as our invocation model.

Next we evaluate the *Predict* operation of μSI. Given the current time t and the elapsed time Δt, to predict the heat field over all the (active or inactive) interactions at $(t+\Delta t)$, the baseline approach is to faithfully simulate the estimating/sampling/updating procedure at each time-tick, which provides the best possible estimation regarding the future state $\mathbf{f}(t+\Delta t)$, featuring linear complexity in terms of Δt. We propose a novel prediction model that trades the prediction accuracy for the execution efficiency. We achieve this by focusing on interactions with higher invocation likelihood (above a threshold h) while ignoring less likely ones.

This prediction accuracy-execution efficiency tradeoff is illustrated in Figure 9. Here the accuracy is measured by the cosine distance between the predicted heat field vector $\hat{\mathbf{f}}(t+\Delta t)$ and the tracked result $\mathbf{f}(t+\Delta t)$. We compare the prediction accuracy and execution time for the baseline simulation approach and the lazy-probing approach (with three different settings of threshold h). Clearly, under acceptable accuracy loss (less than 0.1), the lazy-probing approach achieves considerable saving on execution time, e.g., about 37% for $h = e^{-5}$ and 57% for $h = e^{0}$. However, as h increases, the accuracy loss becomes significant, especially when Δt is large, the error is accumulated at

[6] In following, all the heat intensity measures are in logarithmic scale, unless otherwise noted.

[7] http://snap.stanford.edu

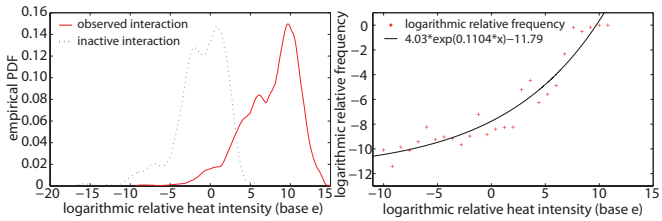

Figure 8: Relative frequency of observed interactions with respect to (relative) heat itensity.

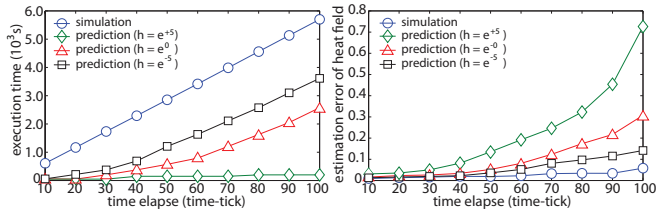

Figure 9: Trade-off between prediction accuracy and execution time.

each time-tick. It is thus important to properly tune h to achieve the balance, e.g., $h = e^0$ in the case here.

5.2 Case Study 2: Paging Operation in Mobile Phone Call Service

For incoming mobile service requests (e.g., voice, SMS, data), efficiently locating requested mobiles or devices is a critical operation for service providers. This is typically done using a combination of location update (by the mobile) and paging (by the network). The paging operation determines where (i.e., which cells) to search for the mobile given its latest location update. Because paging consumes valuable spectrum and signaling resources and the paging channel is a low bandwidth channel, which can also easily become overwhelmed by denial of service attacks, it is considered critical to decrease the utilization of this signaling channel (i.e., minimize the number of cells to be probed).

Existing work on paging schemes has mostly relied on location management-based approaches (e.g., [13]), i.e., create mobility profiles for mobile users, and predict their current locations based on their latest updates and historical profile. In this paper, we take a different approach: instead of focusing on the mobility patterns of requested mobiles (callee), we analyze the *combined social and geographical characteristics* of both caller and callee. We then use such characteristics to design smart paging schemes that minimize signaling traffics.

More specifically, we consider two networks: the social network formed by mobile users, and the cell network formed by cellular base stations. A successfully connected mobile phone call essentially comprises two interactions: caller - caller's station and callee - callee's station. Clearly, understanding the characteristics of these two interactions is the key for us to design effective paging schemes.

Characteristics and Feasibility

We start with examining the geographical characteristics of caller' and callee's base stations. We differentiate the communications between friends and that between non-friends. The distribution of calls with respect to the hop distance of base stations is illustrated in Figure 10. It is noted that both friend and non-friend contacts show fairly similar patterns: the callers tend to contact the callees geographically close to them (but not too close, e.g., within the same cell), while such likelihood decays almost exponentially as hop distance grows. It seems a natural option to use this geographical characteristics to design paging schemes, e.g., given a call request (caller + caller's station + callee), first search base stations with shorter distance to caller's station.

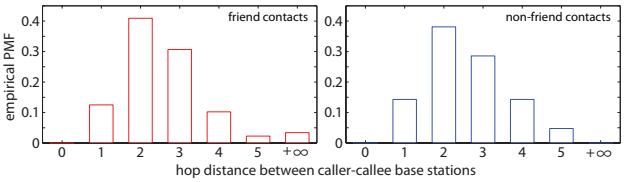

Figure 10: Hop distance between caller's and callee's base stations ($+\infty$ indicates no connection, PMF: probability mass function).

To validate the feasibility of this scheme, and also contrast our solution against traditional mobility profile-based approaches (e.g., [13]), we perform the following experiments. Assume (i) the base station associated with callee as a random variable X, (ii) the callee's historical location logs as: received M calls with the corresponding base stations as a vector $L = (l_1, l_2, \ldots, l_M)$ (K distinct stations in total), each of these K stations appears x_i times in L, $\sum_{i=1}^{K} x_i = M$. We consider the following five measures: (1) entropy of X, entropy of X conditional on (2) previous $N = 1$ station, (3) previous $N = 2$ stations, (4) caller's station, and (5) caller's identify (phone number) and base station. Taking (1) and (2) as an example, the entropy of X is measured as $H(X) = -\sum_{i=1}^{K}(x_i/M)\log(x_i/M)$, while the conditional entropy of X given previous station Y is defined as $H(X,Y) - H(X)$ where $H(X,Y)$ is the entropy of the joint distribution of two consecutive stations in L. Here, scheme (1), (2), and (3) have been used in the state-of-the-art paging schemes [13], while scheme (4) corresponds to the aforementioned caller's station-based scheme. The CDFs of these entropy and conditional entropy are plotted in Figure 11. It is observed: first, the information of caller's station alone may not be sufficient to determine callee's station, while caller's station and caller' and callee' identities together may pinpoint callee's station with fairly high chance; second, this scheme may help improve the traditional mobility profile-based solutions.

Success Rate and Paging Cost

Leveraging the observations in Figure 10 and 11, we construct an influence-based paging scheme. We first modify the cell network: for each station, add links to its hop-2 neighbors, and delete links to hop-1 neighbors

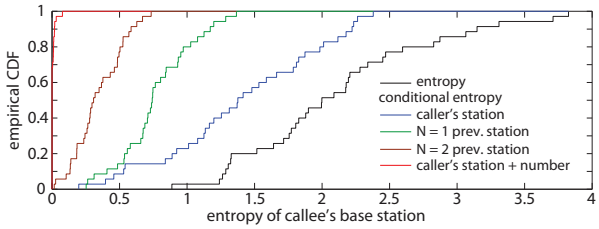

Figure 11: CDF of conditional entropies of callee's base stations.

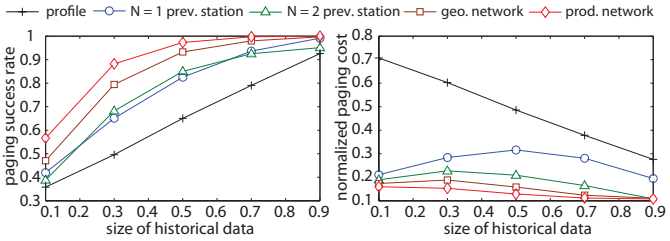

Figure 12: Paging success rate and cost versus historical data size.

(observation in Figure 10); we adopt an instantiation of Tensor product of the cell and social networks. For the set of candidate base stations (maybe returned by profile-based approaches), we sort them according to their heat intensity, and probe the base stations according to this order. Figure 12 shows the efficacy of this paging scheme. We construct five schemes, according to the measures discussed in Figure 11. We measure them using two metrics, the success rate of paging operation, which is defined as the probability that the callee's actual base station appears in the "first-search list" of paging scheme (or a broadcast would be invoked), and the cost of paging, which is defined as the number of base stations actually searched (the results are normalized to $[0, 1]$ where 1 indicates a global broadcast). It is noticed that the influence-based scheme outperforms all the rest in terms of both success rate and paging cost, particularly, when the learning data is limited. This validates our theoretical analysis on its advantage of handling data sparsity issues.

6 CONCLUSION

This work advances the state-of-the-art in modeling the influence between the actions of individuals with social ties. We present a novel heat field over product network model (HFPN) that explicitly accounts for the action-sensitivity, time-sensitivity, and user-sensitivity of such influence. We show that a broad range of applications, such as online resource recommendation and mobile phone call services, can be benefited by this model in terms of accuracy and freshness of customer action prediction.

While the framework of μSI is rich and flexible, several key challenges need to be addressed before it can be readily adopted. (i) *Metrics of influence strength.* To meet our ambitious goal of informative action prediction, it may require to assign influence strength over social ties. Various measures can be used to derive influence strength; it however requires to empirically measure these metrics and identify the optimal one. (ii) *Network evolution.* While our model captures dynamic aspects including the shift of communities' and individuals' interests, as well as the dynamic nature of social influence, it assumes relatively static interconnections between users and objects. How to extend it to support social or object networks in rapid evolution? (iii) *Multi-type objects.* We considered the interactions between users and one type of objects. What if different types of objects are present? How to transfer the knowledge for the interactions with one type of objects to another? Our work might be part of a temporary solution until more comprehensive models are available, and it might inform the design of these models.

Acknowledgements

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. The fourth author is partially supported by grants from NSF CISE NetSE program and NSF Trusted computing program, an IBM SUR grant and an Intel ISTC on Cloud Computing grant.

References

[1] M. Goetz, J. Leskovec, M. McGlohon, and C. Faloutsos, "Modeling blog dynamics," in *ICWSM*, 2009.

[2] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *KDD*, 2003.

[3] S.H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha, "Like like alike - joint friendship and interest propagation in social networks," in *WWW*, 2011.

[4] A. Anagnostopoulos, R. Kumar, and M. Mahdian, "Influence and correlation in social networks," in *KDD*, 2008.

[5] P. Singla and M. Richardson, "Yes, there is a correlation: - from social networks to personal behavior on the web," in *WWW*, 2008.

[6] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *KDD*, 2010.

[7] A. Goyal, F. Bonchi, and L. Lakshmanan, "A data-based approach to social influence maximization," in *VLDB*, 2012.

[8] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *WWW*, 2010.

[9] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *KDD*, 2010.
[10] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi, "Limits of Predictability in Human Mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.
[11] C. A. Hidalgo and C. Rodriguez-Sickert, "The dynamics of a mobile phone network," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 12, pp. 3017 – 3024, 2008.
[12] D. Wang, D. Pedreschi, C. Song, F. Giannotti, A.-L Barabasi, "Human mobility, social ties, and link prediction," in *KDD*, 2011.
[13] H. Zang and J. C. Bolot, "Mining call and mobility data to improve paging efficiency in cellular networks," in *MobiCom*, 2007.
[14] R. Bhatt, V. Chaoji, and R. Parekh, "Predicting product adoption in large-scale social networks," in *CIKM*, 2010.
[15] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *KDD*, 2009.
[16] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *KDD*, 2006.
[17] G. Palla, A. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, pp. 664–667, 2007.
[18] T. Berger-Wolf and J. Saia, "A framework for analysis of dynamic social networks," in *KDD*, 2006.
[19] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *KDD*, 2009.
[20] E. Zheleva, H. Sharara, and L. Getoor, "Co-evolution of social and affiliation networks," in *KDD*, 2009.
[21] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang, "Mining topic-level influence in heterogeneous networks," in *CIKM*, 2010.
[22] P. Cui, F. Wang, S. Liu, M. Ou, and S. Yang. "Who should share what? item-level social influence prediction for users and posts ranking," in *SIGIR*, 2011.
[23] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *ICML*, 2002.
[24] H. Yang, I. King, and M. R. Lyu, "Diffusionrank: a possible penicillin for web spamming," in *SIGIR*, 2007.
[25] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. Mach. Learn. Res.*, vol. 11, pp. 985–1042, 2010.
[26] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: Dynamic tensor analysis," in *KDD*, 2006.
[27] H. Bao and E. Chang, "AdHeat: an influence-based diffusion model for propagating hints to match ads," in *WWW*, 2010.
[28] C.-Y. Lin, N. Cao, S. X. Liu, S. Papadimitriou, J. Sun, and X. Yan, "Smallblue: Social network analysis for expertise search and collective intelligence," in *ICDE*, 2009.
[29] N. Eagle, A. Pentland, and D. Lazer, "Inferring social network structure using mobile phone data," *PNAS*, vol. 106, no. 36, pp. 15 274–15 278, 2009.
[30] J. Peng, D. Zeng, H. Zhao, and F-Y. Wang, "Collaborative filtering in social tagging systems based on joint item-tag recommendations," in *CIKM*, 2010.
[31] Y. Cai, M. Zhang, D. Luo, C. Ding, and S. Chakravarthy, "Low-order tensor decompositions for social tagging recommendation," in *WSDM*, 2011.
[32] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He, "Document recommendation in social tagging services," in *WWW*, 2010.