Learning Social Infectivity in Sparse Low-rank Networks Using Multi-dimensional Hawkes Processes

Ke Zhou
Georgia Institute of Technology

Hongyuan Zha
Georgia Institute of Technology

Le Song
Georgia Institute of Technology

Abstract

How will the behaviors of individuals in a social network be influenced by their neighbors, the authorities and the communities in a quantitative way? Such critical and valuable knowledge is unfortunately not readily accessible and we tend to only observe its manifestation in the form of recurrent and time-stamped events occurring at the individuals involved in the social network. It is an important yet challenging problem to infer the underlying network of social inference based on the temporal patterns of those historical events that we can observe.

In this paper, we propose a convex optimization approach to discover the hidden network of social influence by modeling the recurrent events at different individuals as multi-dimensional Hawkes processes, emphasizing the mutual-excitation nature of the dynamics of event occurrence. Furthermore, our estimation procedure, using nuclear and ℓ_1 norm regularization simultaneously on the parameters, is able to take into account the prior knowledge of the presence of neighbor interaction, authority influence, and community coordination in the social network. To efficiently solve the resulting optimization problem, we also design an algorithm ADM4 which combines techniques of alternating direction method of multipliers and majorization minimization. We experimented with both synthetic and real world data sets, and showed that the proposed method can discover the hidden network more accurately and produce a better predictive model than several baselines.

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31.

1 INTRODUCTION

In today's explosively growing social networks such as Facebook, millions of people interact with each other in a real-time fashion. The decisions made by individuals are largely influenced by their neighbors, the authorities and various communities. For example, a recommendation from a close friend can be very decisive for the purchasing of a product. Thus, the problem of modeling the influences between people is a vital task for studying social networks. Equally important, the issue has also gained much attention in recent years due to its wide-spread applications in e-commerce, online advertisements and so on while the availability of large scale historical social event datasets further fuels its rapid development.

Despite its importance, the network of social influence is usually hidden and not directly measurable. It is different from the "physical" connections in a social network which do not necessarily indicate direct influence — some "friends" in Facebook ("physical" connection) never interact with each other (no influence). However, the network of social influence does manifest itself in the form of various time-stamped and recurrent events occurring at different individuals that are readily observable, and the dynamics of these historical events carry much information about how individuals interact and influence each other. For instance, one might decide to purchase a product the very next day when she saw many people around her bought it today while it may take the same person a prolonged period to make the decision if none from her community has adopted it. Consequently, we set the goal of the paper to address the issue of inferring the network of social influence from the dynamics of these historical events observed on many individuals, and producing a predictive model to answer quantitatively the question: when will this individual take an action given that some other individuals have already done that.

From a modeling perspective, we also want to take into account several key features of social influence. First, many social actions are recurrent in nature. For in-

stance, an individual can participate in a discussion forum and post her opinions multiple times. Second, actions between interacting people are often self- and mutually-exciting. The likelihood of an individual's future participation in an event is increased if she has participated in it before and more so if many of her neighbors also have participated in the event. Third, the network of social influence have certain topological structures. It is usually sparse, *i.e.* most individuals only influence a small number of neighbors, while there are a small number of hubs with wide spread influence on many others. Moreover, people tend to form communities, with the likelihood of taking an action increased under the influence of other members of the same community (assortative, *e.g.,* peer-to-peer relation) or under the influence of members from another specific community (dissortative, *e.g.,* teacher-to-student relation). These topological priors give rise to the structures of the adjacency matrices corresponding to the networks: they tend to have a small number of nonzero entries and they also have sophisticated low-rank structures.

In this paper, we propose a regularized convex optimization approach to discovering the hidden network of social influence based on a multi-dimensional Hawkes process. The multi-dimensional Hawkes process captures the mutually-exciting and recurrent nature of individual behaviors, while the regularization using nuclear norm and ℓ_1 norm simultaneously on the infectivity matrix allows us to impose priors on the network topology (sparsity and low-rank structure). The advantage of our formulation is that the corresponding network discovery problem can be solved efficiently by bringing a large collection of tools developed in the optimization communities. In particular, we developed an algorithm, called ADM4, to solve the problem efficiently by combining the idea of alternating direction method of multipliers [3] and majorization minimization [8]. In our experiments on both synthetic and real world datasets, the proposed method performs significantly better than alternatives in term of accurately discovering the hidden network and predicting the response time of an individual.

2 RELATED WORK

We will start by summarizing related work. Estimating the hidden social influence from historical events is attracting increasing attention recently. For instance, [13] proposes a hidden Markov based model to model the influence between people which treats time as discrete index and hence does not lead to models predictive of the response time. The approach in [6] models the probability of a user influenced by its neighbors by sub-modular functions, but it is not easy to incorporate recurrent events and topological priors in a principled way. In [12, 15], continuous-time models are proposed to recover sparse influence network but the models can not handle recurrent events and they do not take into account the low-rank network structure.

Self-exciting point processes are frequently used to model continuous-time events where the occurrence of one event increases the possibility of future events. Hawkes process [7], an important type of self-exciting process, has been investigated for a wide range of applications such as market modeling [19], earth quake prediction [11], crime modeling [18]. The maximum likelihood estimation of one-dimensional Hawkes process is studied in [10] under the EM framework. Additionally, [16] models cascades of events using marked Poisson processes with marks representing the types of events while [2] propose a model based on Hawkes process that models events between pairs of nodes. The novelty of our paper is the application of multi-dimensional Hawkes process [1, 7] to the problem of discovering hidden influence network, and leverage its connections to convex low-rank matrix factorization techniques.

Low-rank matrix factorizations are applied to a number of real-world problems including collaborative filtering and image processing. Nuclear norm [17] has been shown to be an effective formulation for the estimation of low-rank matrices. On the other hand, ℓ-1 regularization has also been applied to estimate sparse matrices [9]. Furthermore, [14] shows that the accuracy of matrix completion can be improved with both sparsity and low-rank regularizations. One contribution of our paper is to apply these matrix completion results to social influence estimation problem and we also develop a new algorithm ADM4 for solving the corresponding optimization problem efficiently.

3 MULTI-DIMENSIONAL HAWKES PROCESSES WITH LOW-RANK AND SPARSE STRUCTURES

3.1 One-dimensional Hawkes Processes

Before introducing multi-dimensional Hawkes processes, we first describe one-dimensional Hawkes process briefly. In its most basic form, a one-dimensional Hawkes process is a point process N_t with its conditional intensity expressed as follows [7]

$$\lambda(t) = \mu + a \int_{-\infty}^{t} g(t-s) dN_s = \mu + a \sum_{i:t_i<t} g(t-t_i),$$

where $\mu > 0$ is the base intensity, t_i are the time of events in the point process before time t, and $g(t)$ is

the decay kernel. We focus on the case of exponential kernel $g(t) = we^{-wt}$ as a concrete examples in this paper, but the framework discussed in this paper can be easily adapted to other positive kernels. In the above conditional intensity function, the sum over i with $t_i < t$ captures the self-exciting nature of the point process: the occurrence of events in the past has a positive contribution of the event intensity in the future. Given a sequence of events $\{t_i\}_{i=1}^n$ observed in the time interval $[0, T]$ that is generated from the above conditional intensity, the log-likelihood function can be expressed as follows

$$\mathcal{L} = \log \frac{\prod_{i=1}^n \lambda(t_i)}{\exp(\int_0^T \lambda(t)dt)} = \sum_{i=1}^n \log \lambda(t_i) - \int_0^T \lambda(t)dt.$$

3.2 Multi-dimensional Hawkes Processes

In order to model social influence, one-dimensional Hawkes process discussed above needs to be extended to the multi-dimensional case. Specifically, we have U Hawkes processes that are coupled with each other: each of the Hawkes processes corresponds to an individual and the influence between individuals are explicitly modeled. Formally, the multi-dimensional Hawkes process is defined by a U-dimensional point process $N_t^u, u = 1, \ldots, U$, with the conditional intensity for the u-th dimension expressed as follows:

$$\lambda_u(t) = \mu_u + \sum_{i:t_i<t} a_{uu_i} g(t - t_i),$$

where $\mu_u \geq 0$ is the base intensity for the u-th Hawkes process. The coefficient $a_{uu'} \geq 0$ captures the mutually-exciting property between the u-th and u'-th dimension. Intuitively, it captures the degree of influence of events occurred in the u'-th dimension to the u-th dimension. Larger value of $a_{uu'}$ indicates that events in u'-th dimension are more likely to trigger a event in the u-th dimension in the future. We collect the parameters into matrix-vector forms, $\boldsymbol{\mu} = (\mu_u)$ for the base intensity, and $\mathbf{A} = (a_{uu'})$ for the mutually-exciting coefficients, called infectivity matrix. We use $\mathbf{A} \geq 0$ and $\boldsymbol{\mu} \geq 0$ to indicate that we require both matrices to be entry-wise nonnegative.

Suppose we have m samples, $\{c_1, \ldots, c_m\}$, from the multi-dimensional Hawkes process. Each sample c is a sequence of events observed during a time period of $[0, T_c]$, which is in the form of $\{(t_i^c, u_i^c)\}_{i=1}^{n_c}$. Each pair (t_i^c, u_i^c) represents an event occurring at the u_i^c-th dimension at time t_i^c. Thus, the log-likelihood of model parameters $\Theta = \{\mathbf{A}, \boldsymbol{\mu}\}$ can be expressed as follows

$$\begin{aligned}\mathcal{L}(\mathbf{A}, \boldsymbol{\mu}) &= \sum_c \left(\sum_{i=1}^{n_c} \log \lambda_{u_i^c}(t_i^c) - \sum_{u=1}^{U} \int_0^{T_c} \lambda_u(t)dt \right) \\ &= \sum_c \left(\sum_{i=1}^{n_c} \log \left(\mu_{u_i^c} + \sum_{t_j^c<t} a_{u_i^c u_j^c} g(t_i^c - t_j^c) \right) \right. \\ &\quad \left. - T_c \sum_{u=1}^{U} \mu_u - \sum_{u=1}^{U} \sum_{j=1}^{n_c} a_{u u_j^c} G(T_c - t_j^c) \right),\end{aligned}$$

where $G(t) = \int_0^t g(s)ds$. In general, the parameters $\mathbf{A}$ and $\boldsymbol{\mu}$ can be estimated by maximizing the log-likelihood, *i.e.,* $\min_{\mathbf{A}\geq 0, \boldsymbol{\mu}\geq 0} -\mathcal{L}(\mathbf{A}, \boldsymbol{\mu})$.

3.3 Sparse and Low-Rank Regularization

As we mentioned earlier, we would like to take into account the structure of the social influence in the proposed model. We focus on two important properties of the social influences: sparsity and low-rank. The sparsity of social influences implies that most individuals only influence a small fraction of users in the network while there can be a few hubs with wide-spread influence. This can be reflected in the sparsity pattern of $\mathbf{A}$. Furthermore, the communities structure in the influence network implies low-rank structures, which can also be reflected in matrix $\mathbf{A}$. Thus, we consider incorporating this prior knowledge by imposing both low-rank and sparse regularization on $\mathbf{A}$. That is we regularize our maximum likelihood estimator with

$$\min_{\mathbf{A}\geq 0, \boldsymbol{\mu}\geq 0} -\mathcal{L}(\mathbf{A}, \boldsymbol{\mu}) + \lambda_1 \|\mathbf{A}\|_* + \lambda_2 \|\mathbf{A}\|_1, \qquad (1)$$

where $\|\mathbf{A}\|_*$ is the nuclear norm of matrix $\mathbf{A}$, which is defined to be the sum of its singular value $\sum_{i=1}^{\text{rank}\mathbf{A}} \sigma_i$. The nuclear norm has been used to estimate low-rank matrices effectively [17]. Moreover, $\|\mathbf{A}\|_1 = \sum_{u,u'} |a_{uu'}|$ is the ℓ_1 norm of the matrix $\mathbf{A}$, which is used to enforce the sparsity of the matrix $\mathbf{A}$. The parameter λ_1 and λ_2 control the strength of the two regularization terms.

4 EFFICIENT OPTIMIZATION

It can be observed that the objective function in Equation (1) is non-differentiable and thus difficult to optimize in general. We apply the idea of alternating direction method of multipliers (ADMM) [5] to convert the optimization problem to several sub-problems that are easier to solve. The ADMM has been shown to be a special case of the more general Douglas-Rachford splitting method, which has good convergence properties under some fairly mild conditions [4].

Specifically, we first rewrite the optimization problem in Equation (1) to an equivalent form by introducing two auxiliary variables $\mathbf{Z_1}$ and $\mathbf{Z_1}$

$$\min_{\mathbf{A}\geq 0,\boldsymbol{\mu}\geq 0,\mathbf{Z_1},\mathbf{Z_2}} -\mathcal{L}(\mathbf{A},\boldsymbol{\mu})+\lambda_1\|\mathbf{Z_1}\|_*+\lambda_2\|\mathbf{Z_2}\|_1, \quad (2)$$
$$\text{s.t. } \mathbf{A}=\mathbf{Z_1}, \quad \mathbf{A}=\mathbf{Z_2}.$$

In ADMM, we optimize the argumented Lagrangian of the above problem that can be expressed as follows:

$$\begin{aligned}\mathcal{L}_\rho =& -\mathcal{L}(\boldsymbol{\mu},\mathbf{A})+\lambda_1\|\mathbf{Z}_1\|_*+\lambda_2\|\mathbf{Z}_2\|_1\\ &+\rho\text{trace}(\mathbf{U}_1^T(\mathbf{A}-\mathbf{Z}_1))+\rho\text{trace}(\mathbf{U}_2^T(\mathbf{A}-\mathbf{Z}_2))\\ &+\frac{\rho}{2}(\|\mathbf{A}-\mathbf{Z}_1\|^2+\|\mathbf{A}-\mathbf{Z}_2\|^2),\end{aligned}$$

where $\rho>0$ is called the penalty parameter and $\|\cdot\|$ denotes the Frobenius norm. The matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ are the dual variable associated with the constraints $\mathbf{A}=\mathbf{Z}_1$ and $\mathbf{A}=\mathbf{Z}_2$, respectively.

The algorithm for solving the above augmented Lagrangian problem involves the following key iterative steps (also see the details in the Appendix):

$$\mathbf{A}^{k+1},\boldsymbol{\mu}^{k+1}=\operatorname*{argmin}_{\mathbf{A}\geq 0,\boldsymbol{\mu}\geq 0}\mathcal{L}_\rho(\mathbf{A},\boldsymbol{\mu},\mathbf{Z}_1^k,\mathbf{Z}_2^k,\mathbf{U}_1^k,\mathbf{U}_2^k), \quad (3)$$
$$\mathbf{Z}_1^{k+1}=\operatorname*{argmin}_{\mathbf{Z}_1}\mathcal{L}_\rho(\mathbf{A}^{k+1},\boldsymbol{\mu}^{k+1},\mathbf{Z}_1,\mathbf{Z}_2^k,\mathbf{U}_1^k,\mathbf{U}_2^k), \quad (4)$$
$$\mathbf{Z}_2^{k+1}=\operatorname*{argmin}_{\mathbf{Z}_2}\mathcal{L}_\rho(\mathbf{A}^{k+1},\boldsymbol{\mu}^{k+1},\mathbf{Z}_1^{k+1},\mathbf{Z}_2,\mathbf{U}_1^k,\mathbf{U}_2^k), \quad (5)$$
$$\mathbf{U}_1^{k+1}=\mathbf{U}_1^k+(\mathbf{A}^{k+1}-\mathbf{Z}_1^{k+1}),$$
$$\mathbf{U}_2^{k+1}=\mathbf{U}_2^k+(\mathbf{A}^{k+1}-\mathbf{Z}_2^{k+1}).$$

The advantage of sequential update is that we separate multiple variables and thus can optimize them one at a time. We first consider the optimization problem for $\mathbf{Z}_1$ and $\mathbf{Z}_2$ and then describe the algorithm used to optimize with respect to $\mathbf{A}$ and $\boldsymbol{\mu}$.

4.1 Solving for $\mathbf{Z}_1$ and $\mathbf{Z}_2$.

When solving for $\mathbf{Z}_1$ in Equation (4), the relevant terms from $\mathcal{L}_\rho$ are

$$\lambda_1\|\mathbf{Z}_1\|_*+\rho\text{trace}((U_1^k)^T(\mathbf{A}^{k+1}-\mathbf{Z}_1))+\frac{\rho}{2}\|\mathbf{A}^{k+1}-\mathbf{Z}_1\|^2,$$

which can be simplified to an equivalent problem,

$$\mathbf{Z}_1^{k+1}=\operatorname*{argmin}_{\mathbf{Z}_1}\ \lambda_1\|\mathbf{Z}_1\|_*+\frac{\rho}{2}\|\mathbf{A}^{k+1}-\mathbf{Z}_1+\mathbf{U}_1^k\|^2.$$

The above problem has a closed form solution

$$\mathbf{Z}_1^{k+1}=\mathcal{S}_{\lambda_1/\rho}(\mathbf{A}^{k+1}+\mathbf{U}_1^k), \quad (6)$$

where $\mathcal{S}_\alpha(\mathbf{X})$ is a soft-thresholding function defined as $\mathcal{S}_\alpha(\mathbf{X})=\mathbf{U}\text{diag}((\sigma_i-\alpha)_+)\mathbf{V}^T$ for all matrix $\mathbf{X}$ with singular value decomposition $\mathbf{X}=\mathbf{U}\text{diag}(\sigma_i)\mathbf{V}^T$.

Similarly, the optimization for $\mathbf{Z}_2$ can be simplified into the following equivalent form

$$\mathbf{Z}_2^{k+1}=\operatorname*{argmin}_{\mathbf{Z}_2}\ \lambda_2\|\mathbf{Z}_2\|_1+\frac{\rho}{2}\|\mathbf{A}^{k+1}-\mathbf{Z}_2+\mathbf{U}_2^k\|^2,$$

which again has the closed form solution. In this case, depending on the magnitude of the ij-th entry of the matrix $(\mathbf{A}^{k+1}+\mathbf{U}_2^k)$, the corresponding $(\mathbf{Z}_2^{k+1})_{ij}$ is updated as

$$\begin{cases}(\mathbf{A}^{k+1}+\mathbf{U}_2^k)_{ij}-\frac{\lambda_2}{\rho}, & (\mathbf{A}^{k+1}+\mathbf{U}_2^k)_{ij}\geq\frac{\lambda_2}{\rho},\\ (\mathbf{A}^{k+1}+\mathbf{U}_2^k)_{ij}+\frac{\lambda_2}{\rho}, & (\mathbf{A}^{k+1}+\mathbf{U}_2^k)_{ij}\leq-\frac{\lambda_2}{\rho},\\ 0, & |(\mathbf{A}^{k+1}+\mathbf{U}_2^k)_{ij}|<\frac{\lambda_2}{\rho}.\end{cases} \quad (7)$$

4.2 Solving for $\mathbf{A}$ and $\boldsymbol{\mu}$

The optimization problem for $\mathbf{A}$ and $\boldsymbol{\mu}$ defined in Equation (3) can be equivalently written as

$$\mathbf{A}^{k+1},\boldsymbol{\mu}^{k+1}=\operatorname*{argmin}_{\mathbf{A}\geq 0,\boldsymbol{\mu}\geq 0}\ f(\mathbf{A},\boldsymbol{\mu})$$

where $f(\mathbf{A},\boldsymbol{\mu})=-\mathcal{L}(\mathbf{A},\boldsymbol{\mu})+\frac{\rho}{2}(\|\mathbf{A}-\mathbf{Z}_1^k+\mathbf{U}_1^k\|^2+\|\mathbf{A}-\mathbf{Z}_2^k+\mathbf{U}_2^k\|^2)$. We propose to solve the above problem by a majorization-minimization algorithm which is a generalization of the EM algorithm. Since the optimization is convex, we still obtain global optimum for this subproblem. Specifically, given any estimation $\mathbf{A}^{(m)}$ and $\boldsymbol{\mu}^{(m)}$ of $\mathbf{A}$ and $\boldsymbol{\mu}$, we minimize a surrogate function $Q(\mathbf{A},\boldsymbol{\mu};\mathbf{A}^{(m)},\boldsymbol{\mu}^{(m)})$ which is a tight upper bound of $f(\mathbf{A},\boldsymbol{\mu})$. Indeed, $Q(\mathbf{A},\boldsymbol{\mu};\mathbf{A}^{(m)},\boldsymbol{\mu}^{(m)})$ can be defined as follows:

$$\begin{aligned}&Q(\mathbf{A},\boldsymbol{\mu};\mathbf{A}^{(m)},\boldsymbol{\mu}^{(m)})\\ =&-\sum_c\left(\sum_{i=1}^{n^c}\left(p_{ii}^c\log\frac{\mu_{u_i^c}}{p_{ii}^c}+\sum_{j=1}^{i-1}p_{ij}^c\log\frac{a_{u_i^cu_j^c}g(t_i^c-t_j)}{p_{ij}^c}\right)\right.\\ &\left.-\left(T_c\sum_u\mu_u+\sum_{u=1}^U\sum_{j=1}^{n^c}a_{uu_j^c}G(T-t_j^c)\right)\right)\\ &+\frac{\rho}{2}(\|\mathbf{A}-\mathbf{Z}_1^k+\mathbf{U}_1^k\|^2+\|\mathbf{A}-\mathbf{Z}_2^k+\mathbf{U}_2^k\|^2), \quad (8)\end{aligned}$$

where

$$p_{ii}^c=\frac{\mu_{u_i^c}^{(m)}}{\mu_{u_i^c}^{(m)}+\sum_{j=1}^{i-1}a_{u_i^cu_j^c}^{(m)}g(t_i^c-t_j^c)},$$
$$p_{ij}^c=\frac{a_{u_i^cu_j^c}^{(m)}g(t_i^c-t_j^c)}{\mu_{u_i^c}^{(m)}+\sum_{j=1}^{i-1}a_{u_i^cu_j^c}^{(m)}g(t_i^c-t_j^c)}.$$

Intuitively, p_{ij}^c can be interpreted as the probability that the i-th event is influenced by a previous event j in the network and p_{ii}^c is the probability that i-th event is sampled from the base intensity. Thus, the first

Algorithm 1 ADMM-MM (ADM4) for estimating $\mathbf{A}$ and $\boldsymbol{\mu}$

Input: Observed samples $\{c_1, \ldots, c_m\}$.
Output: $\mathbf{A}$ and $\boldsymbol{\mu}$.
Initialize $\boldsymbol{\mu}$ and $\mathbf{A}$ randomly; Set $\mathbf{U}_1 = 0$, $\mathbf{U}_2 = 0$.
while $k = 1, 2, \ldots,$ **do**
Update $\mathbf{A}^{k+1}$ and $\boldsymbol{\mu}^{k+1}$ by optimizing Q defined in (8) as follows:
while not converge **do**
Update $\mathbf{A}$, $\boldsymbol{\mu}$ using (10) and (9) respectively.
end while
Update $\mathbf{Z}_1^{k+1}$ using (6); Update $\mathbf{Z}_2^{k+1}$ using (7).
Update $\mathbf{U}_1^{k+1} = \mathbf{U}_1^k + (\mathbf{A}^{k+1} - \mathbf{Z}_1^{k+1})$ and $\mathbf{U}_2^{k+1} = \mathbf{U}_2^k + (\mathbf{A}^{k+1} - \mathbf{Z}_2^{k+1})$.
end while
return $\mathbf{A}$ and $\boldsymbol{\mu}$.

two terms of $Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)})$ can be viewed as the joint probability of the unknown infectivity structures and the observed events.

As is further shown in the Appendix, optimizing $Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)})$ ensures that $f(\mathbf{A}, \boldsymbol{\mu})$ is decreasing monotonically. Moreover, the advantage of optimizing $Q(\mathbf{A}, \boldsymbol{\mu})$ is that all parameters $\mathbf{A}$ and $\boldsymbol{\mu}$ can be solved independently with each other with closed forms solutions, and the nonnegativity constraints are automatically taken care of. That is

$$\mu_u^{(m+1)} = \frac{\sum_c \sum_{i:i\leq n^c, u_i^c = u} p_{ii}^c}{\sum_c T_c} \tag{9}$$

$$a_{uu'}^{(m+1)} = \frac{-B + \sqrt{B^2 + 8\rho C}}{4\rho}, \tag{10}$$

where

$$B = \sum_c \sum_{j:u_j^c = u'} (G(T - t_j^c)) + \rho(-z_{1,uu'} + u_{1,uu'} - z_{2,uu'} + u_{2,uu'}),$$

$$C = \sum_c \sum_{i=1, u_i^c = u}^{n^c} \sum_{j<i, u_j^c = u'} p_{ij}^c.$$

The overall optimization algorithm is summarized in Algorithm 1.

5 EXPERIMENTS

In this section, we conducted experiments on both synthetic and real-world datasets to evaluate the performance of the proposed method.

5.1 Synthetic Data

Data Generation. The goal is to show that our proposed algorithm can reconstruct the underlying parameters from observed recurrent events. To this end, we consider a U-dimensional Hawkes process with $U = 1000$ and generate the true parameters $\boldsymbol{\mu}$ from a uniform distribution on $[0, 0.001]$. In particular, the infectivity matrix $\mathbf{A}$ is generated by $\mathbf{A} = \mathbf{U}\mathbf{V}^T$. We consider two different types of influences in our experiments: assortative mixing and disassortative mixing:

- In the assortative mixing case, $\mathbf{U}$ and $\mathbf{V}$ are both 1000×9 matrices with entries in $[100(i-1)+1 : 100(i+1), i], i = 1, \ldots, 9$ sampled randomly from $[0, 0.1]$ and all other entries are set to zero. Assortative mixing examples capture the scenario that influence are mostly coming from members of the same group.
- In the disassortative mixing case, $\mathbf{U}$ is generated in the same way as the assortative mixing case, while the $\mathbf{V}$ has non-zero entries in $[100(i-1)+1 : 100(i+1), 10-i], i = 1, \ldots, 9$. Disassortative mixing examples capture the scenario that influence can come outside of the group, possibly from a few influential hubs.

We scale $\mathbf{A}$ so that the spectral radius of $\mathbf{A}$ is 0.8 to ensure the point process is well-defined, i.e., with finite intensity[1]. Then, 50000 samples are sampled from the multi-dimensional Hawkes process specified by $\mathbf{A}$ and $\boldsymbol{\mu}$. The proposed algorithms are applied to the samples to obtain estimations $\hat{\mathbf{A}}$ and $\hat{\boldsymbol{\mu}}$.

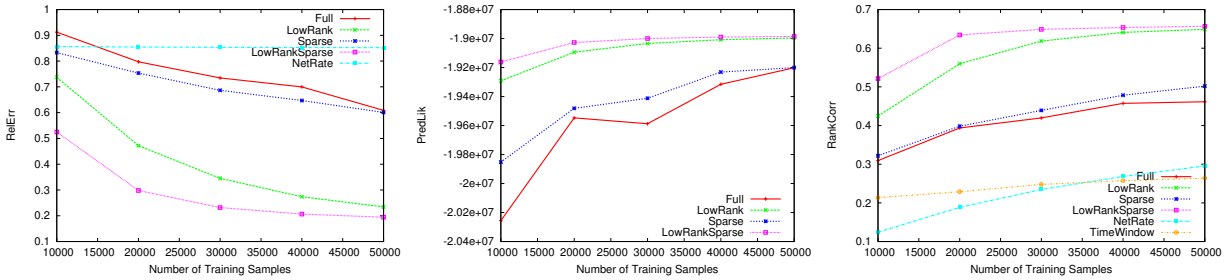
Evaluation Metric. We use three evaluation metrics to measure the performance:

- RelErr is defined as the averaged relative error between the estimated parameters and the true parameters, *i.e.* $\frac{|a_{ij} - \hat{a}_{ij}|}{|a_{ij}|}$ for $a_{ij} \neq 0$ and $|a_{ij} - \hat{a}_{ij}|$ for $a_{ij} = 0$.
- PredLik is defined as the log-likelihood of the estimated model on a separate held-out test set containing 50,000 samples.
- RankCorr is defined as the averaged Kendall's rank correlation coefficient between each row of $\mathbf{A}$ and $\hat{\mathbf{A}}$. It measures whether the relative order of the estimated social influences is correctly recovered.

Results. We included 5 methods in the comparisons:

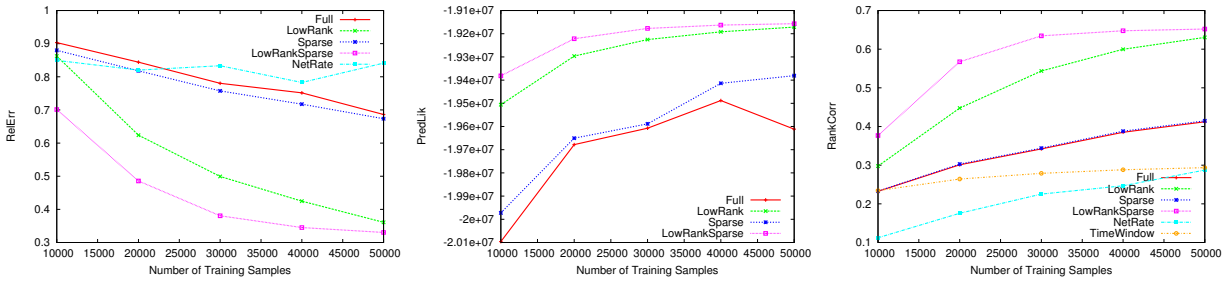
- TimeWindow. This method, we first discretize time into time windows with equal length and then represent each node by a vector where each dimension is to the number of events occurred within the corresponding time window at the

[1] In our case, the point process is well-defined iff the specitral radius of $\mathbf{A}$ is less than 1



(a) RelErr (b) PredLik (c) RankCorr

Figure 1: Assortative mixing networks: performance measured by RelErr, PredLik and RankCorr with respect to the number of training samples.



(a) RelErr (b) PredLik (c) RankCorr

Figure 2: Disassortative mixing networks: Performance measured by RelErr, PredLik and RankCorr with respect to the number of training samples.

node. The cosine similarity are used to estimate the infectivity matrix.

- NetRate. This method is proposed in [15] for modeling information diffusion in networks. It can not model the recurrent events, so we only keep the first occurrences at each node in the training data.
- Full. The infectivity matrix $\mathbf{A}$ is estimated as a general $U \times U$ matrix without any structure.
- LowRank. Only the nuclear norm is used to obtain a low-rank estimation of $\mathbf{A}$.
- Sparse. Only the ℓ_1 norm is used to obtain a sparse estimation of $\mathbf{A}$.
- LowRankSparse. This is the proposed method Algorithm 1. Both nuclear norm and ℓ_1 norm are used to estimate $\mathbf{A}$.

For each method, the parameters are selected on a validation set that are disjoint from both training and test set. We run each experiment for five times with different samples and report the averaged performance metrics over all the five runs.

Figure 1 plots the results on assortative mixing networks measured by RelErr, PredLik and RankCorr with respect to the number of training data. It can be observed from Figure 1 that when the number of training samples increase, the RelErr decreases and both PredLik and RankCorr increase, indicating that all methods can improve accuracy of estimation with more training samples. Moreover, LowRank and Sparse outperforms Full in all cases. Therefore, we conclude that utilizing the structure of the matrix can improve the estimation very significantly. LowRankSparse outperforms all other baselines since it fully utilizes prior information about the infectivity matrix. It is interesting to observe that when the number of training samples are small, the improvements of LowRankSparse over other baselines are very large. We think this is because when the number of training samples are not sufficiently large to get a good estimation, the prior knowledge from the structure of the infectivity matrix becomes more important and useful. TimeWindow is not as good as other methods since it can not capture the time pattern very accurately. Similarly in Figure 2, the proposed method LowRankSparse outperforms other methods in disassortative mixing networks. These two sets of experiments indicate that LowRankSparse can handle well different network topologies.

In Figure 3 and Figure 4, we plot the performance with respect to the values of the two parameters λ_1 and λ_2 in LowRankSparse. It can be observed that the performance first increases and then decreases when the value λ_1 grows. Note that when $\lambda_1 = 0$, we obtain

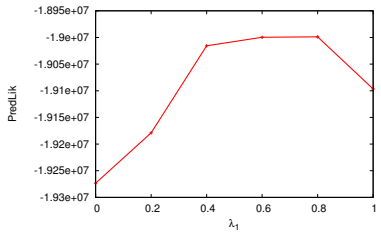


Figure 3: Performance measured by PredLik with respect to the value of λ_1.

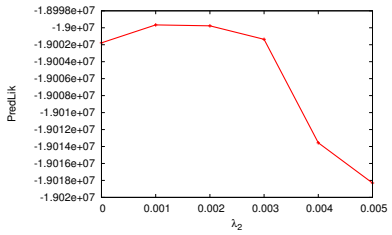


Figure 4: Performance measured by PredLik with respect to the value of λ_2.

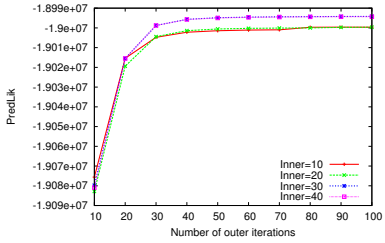


Figure 5: Performance measured by PredLik with respect to the number of inner/outer iterations.

a sparse solution that is not low-rank, which is underperformed by solutions that are both low-rank and sparse. Similar observations can be made for λ_2. In general, we use $\lambda_2 = 0.6$,$\lambda_1 = 0.02$ for those Figure 3 and Figure 4, respectively.

In order to investigate the convergence of the proposed algorithm, in Figure 5, we present the performance measured by PredLik with respect to the number of outer iterations K in Algorithm 1. We can observe that the performance measured by PredLik grows with the number of outer iterations and converges within about 50 iterations. We also illustrate the impact of the number of inner iterations in Figure 5. It can be observed that larger number of inner iterations leads to better convergence speed and slightly better performance.

5.2 Real-world Data

We also evaluate the proposed method on a real world data set. To this end, we use the MemeTracker data set[2]. The data set contains the information flows captured by hyper-links between different sites with timestamps. In particular, we first extract the top 500 popular sites and the links between them. The events are in the form that a site created a hyper-link to another site at a particular time. We use 50% data as training data and 50% as test data.

In Figure 6, we show that negative log-likelihood of Full, Sparse, LowRank and LowRankSparse on the test set. We can see that LowRankSparse outperforms the baselines. Therefore, we conclude that LowRankSparse can better model the influences in social networks.

We also study whether the proposed model can discover the influence network between users from the recurrent events. To this end, we present the RankCorr of Full, Sparse, LowRank and LowRankSparse in Figure 7. We also include NetRate as a baseline. It can be observed that the LowRankSparse obtains better rank correlation than other models, which indicates that it can capture the influence network better than other models. In Figure 8, we visualize the influence network estimated from the MemeTracker data. We can observe that there is a quite dense region near the bottom right in the infectivity matrix. This region represents that the corresponding sites are the centric of the infectivity networks. Example sites in this region include `news.cnet.com`, `blogs.zdnet.com` and `blogs.abcnews.com`. The first two are both famous IT news portal and the third one is a blog site that belongs to a general news portal. It is clear that all of these sites are popular sites that can quickly detect trending events and propagate them to a lot of other sites.

[2] `http://memetracker.org`

6 CONCLUSIONS

In this paper, we propose to infer the network social influence from the observed recurrent events indicating users' activities in the social networks. The proposed model utilizes the mutually-exciting multi-dimensional Hawkes model to capture the temporal patterns of user behaviors. Moreover, we estimate the infectivity matrix for the network that is both low-rank and sparse by optimizing nuclear norm and ℓ_1 norm simultaneously. The resulting optimization problem is solved through combining the ideas of alternating direction method of multipliers and majorization-minimization. The experimental results on both simulation and real-world datasets suggest that the proposed model can estimate the social influence between users accurately.

There are several interesting directions for future studies: First, we plan to investigate the adaptation of the proposed model to the problem of collaborative filtering. In particular, we plan to model the social influences to capture both short-term and long-term preferences of users. Moreover, we can also investigate the problem of estimating the decay kernel together with other parameters in the model.

Acknowledgements
Part of the work is supported by NSF IIS-1116886, NSF IIS-1218749 and a DARPA Xdata grant.

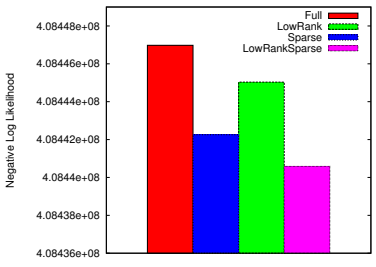


Figure 6: Performance measured by PredLik on MemeTracker data set.

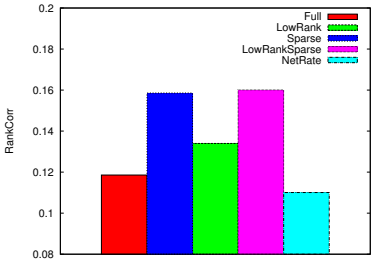


Figure 7: Performance measured by RankCorr on MemeTracker data set.

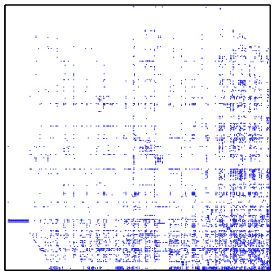


Figure 8: Influence structure estimated from the MemeTracker data set.

Appendix

Derivation of ADMM. In ADMM, we consider the *argumented Lagrangian* of the above constrained optimization problem by writing it as follows:

$$\min -\mathcal{L}(\boldsymbol{\mu}, \mathbf{A}) + \lambda_1 \|\mathbf{Z}_1\|_* + \lambda_2 \|\mathbf{Z}_2\|_1 + \frac{\rho}{2}(\|\mathbf{A}-\mathbf{Z}_1\|^2 + \|\mathbf{A}-\mathbf{Z}_2\|^2) \quad (11)$$

subject to $\mathbf{A} = \mathbf{Z}_1, \mathbf{A} = \mathbf{Z}_2$. Clearly, for all ρ the optimization problem defined above is equivalent to Equation (2) and thus equivalent to the problem defined in Equation (1). The argumented Lagrangian of Equation (2) is the (standard) Lagrangian of Equation (11), which can be expressed as follows:

$$\begin{aligned}\mathcal{L}_\rho = &-\mathcal{L}(\boldsymbol{\mu}, \mathbf{A}) + \lambda_1 \|\mathbf{Z}_1\|_* + \lambda_2 \|\mathbf{Z}_2\|_1 \\ &+ \operatorname{trace}(\mathbf{Y}_1^T(\mathbf{A}-\mathbf{Z}_1)) + \operatorname{trace}(\mathbf{Y}_2^T(\mathbf{A}-\mathbf{Z}_2)) \\ &+ \frac{\rho}{2}(\|\mathbf{A}-\mathbf{Z}_1\|^2 + \|\mathbf{A}-\mathbf{Z}_2\|^2)\end{aligned}$$

Thus, we can solve the optimization problem defined in Equation (2) applying the gradient ascent algorithm to the dual variables $\mathbf{Y}_1$ and $\mathbf{Y}_2$. It can be shown that the update of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ has the following form at the k-th iteration:

$$\begin{aligned}\mathbf{Y}_1^{k+1} &= \mathbf{Y}_1^k + \rho(\mathbf{A}^{k+1} - \mathbf{Z}_1^{k+1}) \\ \mathbf{Y}_2^{k+1} &= \mathbf{Y}_2^k + \rho(\mathbf{A}^{k+1} - \mathbf{Z}_2^{k+1}),\end{aligned}$$

where $\mathbf{A}^{k+1}$, $\mathbf{Z}_1^{k+1}$ and $\mathbf{Z}_2^{k+1}$ are obtained by optimizing $\mathcal{L}_\rho$ with $\mathbf{Y}_1 = \mathbf{Y}_1^k$ and $\mathbf{Y}_2 = \mathbf{Y}_2^k$ fixed:

$$\operatorname*{argmin}_{\mathbf{A},\boldsymbol{\mu},\mathbf{Z}_1,\mathbf{Z}_2} \mathcal{L}_\rho(\mathbf{A}, \boldsymbol{\mu}, \mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Y}_1^k, \mathbf{Y}_2^k)$$

In ADMM, the above problem is solved by updating $\mathbf{A}$, $\boldsymbol{\mu}$, $\mathbf{Z}_1$ and $\mathbf{Z}_2$ sequentially as follows:

$$\begin{aligned}\mathbf{A}^{k+1}, \boldsymbol{\mu}^{k+1} &= \operatorname*{argmin}_{\mathbf{A},\boldsymbol{\mu},} \mathcal{L}_\rho(\mathbf{A}, \boldsymbol{\mu}, \mathbf{Z}_1^k, \mathbf{Z}_2^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k), \\ \mathbf{Z}_1^{k+1} &= \operatorname*{argmin}_{\mathbf{Z}_1} \mathcal{L}_\rho(\mathbf{A}^{k+1}, \boldsymbol{\mu}^{k+1}, \mathbf{Z}_1, \mathbf{Z}_2^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k), \\ \mathbf{Z}_2^{k+1} &= \operatorname*{argmin}_{\mathbf{Z}_2} \mathcal{L}_\rho(\mathbf{A}^{k+1}, \boldsymbol{\mu}^{k+1}, \mathbf{Z}_1^{k+1}, \mathbf{Z}_2, \mathbf{Y}_1^k, \mathbf{Y}_2^k).\end{aligned}$$

It is usually more convenient to consider the scaled form of ADMM. Let $\mathbf{U}_1^k = \mathbf{Y}_1^k/\rho$ and $\mathbf{U}_2^k = \mathbf{Y}_2^k/\rho$, we obtain the algorithm described in Section 4.

Majorization Minmization. First, we claim that the following properties hold for $Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)};)$ defined in Equation (8) :

1. For all $\mathbf{A}$, $\boldsymbol{\mu}$,

$$Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \geq f(\mathbf{A}, \boldsymbol{\mu})$$

2.

$$Q(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) = f(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)})$$

Proof. The first claim can be shown by utilizing the Jensen's inequality: For all c and i, we have

$$\log(\mu_{u_i^c} + \sum_{j=1}^{i-1} a_{u_i^c u_j^c} g(t_i^c - t_j^c)) \geq p_{ii}^c \log \frac{u_i^c}{p_{ii}^c} + \sum_{j=1}^{i-1} p_{ij}^c \frac{a_{u_i^c u_j^c} g(t_i^c - t_j^c))}{p_{ij}^c}$$

Summing up over c and i proves the claim.

The second claim can be checked by setting $\mathbf{A} = \mathbf{A}^{(m)}$ and $\boldsymbol{\mu} = \boldsymbol{\mu}^{(m)}$. □

The above two properties imply that if $(\mathbf{A}^{(m+1)}, \boldsymbol{\mu}^{(m+1)}) = \operatorname{argmin}_{\mathbf{A},\boldsymbol{\mu}} Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)})$, we have

$$\begin{aligned}f(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) &= Q(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \\ &\geq Q(\mathbf{A}^{(m+1)}, \boldsymbol{\mu}^{(m+1)}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \\ &\geq f(\mathbf{A}^{(m+1)}, \boldsymbol{\mu}^{(m+1)}).\end{aligned}$$

Thus, optimizing Q with respect to $\mathbf{A}$ and $\boldsymbol{\mu}$ ensures that the value of $f(\mathbf{A}, \boldsymbol{\mu})$ decrease monotonically.

References

[1] L. Adamopoulos. Some counting and interval properties of the mutually-exciting processes.

Journal of Applied Probability, 12(1):78, Mar. 1975.

[2] C. Blundell, K. Heller, and J. Beck. Modelling reciprocating relationships with Hawkes processes. *NIPS*, 2012.

[3] S. Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[4] J. Eckstein and D. P. Bertsekas. On the Douglas Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, Apr. 1992.

[5] D. Gabay and B. Mercier. A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation. *Computers & Mathematics with Applications*, 2(1):17–40, Jan. 1976.

[6] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining - WSDM '10*, page 241, New York, New York, USA, 2010. ACM Press.

[7] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[8] D. R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, (58), 2004.

[9] M. Kolar, L. Song, A. Ahmed, and E. Xing. Estimating time-varying networks. *The Annals of Applied Statistics*, 4(1):94–123, 2010.

[10] E. Lewisa and G. Mohlerb. A Nonparametric EM algorithm for multiscale Hawkes processes. *Journal of Nonparametric Statistics*, (1), 2011.

[11] D. Marsan and O. Lengliné. Extending earthquakes' reach through cascading. *Science (New York, N.Y.)*, 319(5866):1076–9, Feb. 2008.

[12] S. Myers and J. Leskovec. On the convexity of latent social network inference. *NIPS*, 2010.

[13] W. Pan, M. Cebrian, W. Dong, T. Kim, J. Fowler, and A. Pentland. Modeling dynamical influence in human interaction patterns. *Arxiv preprint arXiv:1009.0240*, page 19, Sept. 2010.

[14] E. Richard, P.-A. Savalle, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. *ICML*, June 2012.

[15] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. *Proceedings of the 28th International Conference on Machine Learning*, pages 561–568, May 2011.

[16] A. Simma and M. Jordan. Modeling events with cascades of Poisson processes. *UAI*, 2012.

[17] N. Srebro. Learning with matrix factorizations. *Thesis*, 2004.

[18] A. Stomakhin, M. B. Short, and A. L. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 27(11):115013, Nov. 2011.

[19] I. M. Toke. "Market making" behaviour in an order book model and its impact on the bid-ask spread. *Arxiv*, page 17, Mar. 2010.