Multivariate Hawkes Processes for Large-scale Inference

Rémi Lemonnier[1,2] **Kevin Scaman**[1,3] **Argyris Kalogeratos**[1]
[1] CMLA – ENS Cachan, CNRS, Université Paris-Saclay, France
[2] Numberly, 1000Mercis group, Paris, France
[3] Microsoft–Inria Joint Center, Paris, France
{lemonnier, scaman, kalogeratos}@cmla.ens-cachan.fr

Abstract

In this paper, we present a framework for fitting multivariate Hawkes processes for large-scale problems, both in the number of events in the observed history n and the number of event types d (i.e. dimensions). The proposed *Scalable Low-Rank Hawkes Process* (SLRHP) framework introduces a low-rank approximation of the kernel matrix that allows to perform the nonparametric learning of the d^2 triggering kernels in at most $O(ndr^2)$ operations, where r is the rank of the approximation ($r \ll d, n$). This comes as a major improvement to the existing state-of-the-art inference algorithms that require $O(nd^2)$ operations. Furthermore, the low-rank approximation allows SLRHP to learn representative patterns of interaction between event types, which is usually valuable for the analysis of complex processes in real-world networks.

Introduction

In many real-world phenomena, such as product adoption or information sharing, events exhibit a *mutually-exciting* behavior, in the sense that the occurrence of one event can increase the occurrence rate of other events. In the field of internet marketing, a client's purchasing behavior on one online shopping website can be, to a large extent, predicted by his past navigation history on other websites. In finance, arrivals of buying and selling orders for different stocks convey information about macroscopic market tendencies. In the study of information propagation, users of a social network share information, which leads to *information cascades* spreading throughout the social graph. Over the past few years, the study of point processes gained attention as the acquisition of such datasets by companies and research laboratories became simpler. However, the traditional models for time series analysis, such as discrete-time autoregressive models, do not apply in this context due to the fact that events happen in a continuous way.

Multivariate Hawkes processes (MHP) [1, 2] have emerged in several fields as the gold standard to deal with such data, e.g. earthquake prediction [3], biology [4], financial [5, 6], and social interactions studies [7]. For MHP, an event of type u (e.g. a visit to a product's website) occurring at time t, will increase the conditional occurrence rate of events of type v at time $s \geq t$ (e.g. purchases of that product in the future) by a rate $g_{uv}(s-t)$. Despite these processes have been extensively studied from the probabilistic point of view (stability [8], cluster representation [9]), their application to real-scale datasets remains quite challenging. For instance, social interactions data is at the same time *big* (large number of posts), *high-dimensional* (large number of users), and *structured* (social network).

Several nonparametric estimation procedures have been proposed for MHP. relying on approaches such as moment matching [10, 11], least-squares error minimization [12], or log-likelihood maximization [13]. Regardless to the approach each of those works adopts, the dependence of the stochastic occurrence rate at a given time on all past occurrences, and the fact that all d^2 triggering kernels g_{uv} need to be estimated, do imply that all these methods are quadratic in the number of events n as well as the number of dimensions d. Therefore, they quite impractical for large datasets. Subsequent works aimed naturally at increasing scalability. In order to reduce complexity towards $O(n)$, a nonparametric estimation procedure linear in the number of events was proposed in [14], relying on the *memoryless property* of Hawkes processes with exponential triggering kernels, thus achieving an overall complexity of $O(nd^2)$. Moreover, several other works [15, 16, 17] managed a complexity in $O(n^2 d)$ by imposing a low-rank structure to the amplitude of the mutual excitation, while keeping a fixed temporal excitation pattern. Although these methods may exhibit a linear complexity in d, they only impose a community structure to the network via a low-rank assumption on the adjacency matrix, instead of learning an excitation function for each group independently. Note that this difference is significant since learning independent excitation functions allows to uncover groups of users sharing a similar "role" (e.g. influencer vs influencee), instead of mere clusters of densely connected nodes in the network.

In this paper we introduce the *Scalable Low-Rank Hawkes Processes* (SLRHP) model for structured point processes, relying on a *low-rank decomposition of the triggering kernel*, that aims to learn representative patterns of interaction between event types. We also provide the first inference algorithm for SLRHP that has *linear complexity in the total number of events and number of event types* (i.e. dimensions), that is $O(nd)$. The inference is performed by combin-

ing minorize-maximization and self-concordant optimization techniques. In addition, if the underlying network of interactions is provided, then SLRHP is capable of fully exploiting the network sparsity, which renders it practical for large structured datasets. The major advantage of the proposed SLRHP algorithm is the ability to scale-up to datasets much larger than existing state-of-the-art methods, while achieving performance very close to state-of-the-art competitors (in terms of prediction and inference accuracy) on synthetic as well as real datasets.

Setup and Notations

A multivariate Hawkes process (MHP) $N(t) = \{N_u(t) : u = 1, ..., d, t \geq 0\}$ is a d-dimensional counting process, where $N_u(t)$ is the number of events along dimension u which occurred during time $[0, t]$. We call as *event of type* u an event that occurs along dimension u. Each one-dimensional counting process $N_u(t)$ can be influenced by the occurrence of events of other types. Without loss of generality, we consider that these *mutual excitations* take place along the edges of an unweighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of d nodes and adjacency matrix $A \in \{0, 1\}^{d \times d}$. Note that this setting includes in particular the standard definition of multivariate Hawkes processes. Finally, let $\mathcal{H} : (u_m, t_m)_{m=1}^n$ be the event history of the process indicating for each event m its type u_m and occurrence time t_m. The non-negative stochastic occurrence rate of each $N_u(t)$ is then defined by:

$$\lambda_u(t) = \mu_u(t) + \sum_{m:t_m<t} A_{u_m u}\, g_{u_m u}(t - t_m). \qquad (1)$$

In the above, $\mu_u(t) \geq 0$ is the *natural occurrence rate* of events of type u (i.e. along that dimension) at time t, and the *triggering kernel function* evaluation $g_{vu}(s - t) \geq 0$ determines the *increase* in the occurrence rate of events of type u at time s, caused by an event of type v at a past time $t \leq s$.

The natural occurrence rates μ_u and triggering kernels g_{vu} are usually inferred by means of log-likelihood maximization. The main practical issue for inferring the parameters of the model in Eq. 1 is that it requires a particularly large dataset of observations, as standard inference algorithms require at least one observation per pair of event types (i.e. d^2 observations).

In many practical situations, the underlying network of interactions is unknown. In this case, the model to use will be a standard multivariate Hawkes process, which corresponds to taking $A_{uv} = 1$ for every pair of event types (u, v), and the inference procedure proposed in this paper will discover by itself which interactions are non-negligible. However, the reason for which we use an adjacency matrix in our definition is that we show in the following that our model can take advantage of additional information on the support of interactions if provided. In any case, we point out that the applicability of our method is not conditioned on previous knowledge of the support of interactions and that we do not aim to perform network inference, as in [18] whose goal is to learn a sparse and low-rank support of interactions given a parametric form of the triggering kernels.

Scalable Low-Rank Hawkes Processes

The proposed model

Model considerations. Standard MHP inference requires the learning of d^2 triggering kernels that encode the cross- and self-excitement of the event types. Apparently, it becomes prohibitive to satisfy this requirement as d gets larger (e.g. when the dimensions represent the users of a social network, or websites on the Internet). However, in a number of practical situations, the d^2 complex interactions between event types can be summarized by considering that there is a small number of r *event groups* and each event type is related to each of those groups to a certain extent. Therefore, one needs to simultaneously learn a $d \times r$ event type-to-group(s) mapping (we specifically use *soft* assignments) as well as the r^2 interactions between pairs of event groups.

Model formulation. *Scalable Low-Rank Hawkes Processes* (SLRHP) simplify the standard inference process by projecting the original d event types (i.e. dimensions) of a multivariate Hawkes process into a smaller and more compact r-dimensional space. The natural occurrence rates μ_u and triggering kernels g_{vu} of Eq. 1 are then defined via the low-rank approximation:

$$\begin{aligned} \mu_u(t) &= \textstyle\sum_{i=1}^r P_{ui}\, \tilde{\mu}_i(t); \\ g_{vu}(t) &= \textstyle\sum_{i,j=1}^r P_{ui}\, P_{vj}\, \tilde{g}_{ji}(t), \end{aligned} \qquad (2)$$

where u, v are event types, $P \in \mathbb{R}_+^{d \times r}$ is the projection matrix from the original d-dimensional space to the low-dimensional space, and i, j are its component directions. Besides, this projection can be seen as a low-rank approximation of the kernel matrix g since, in matrix notations, $g = P \tilde{g} P^\top$ and $\tilde{g} \in \mathbb{R}_+^{r \times r}$ is a matrix of size $r \ll d$.

Then, the SLRHP occurrence rates are formulated as an extension of Eq. 1 that uses an embedding of event types in a low-dimensional space:

$$\begin{aligned} \lambda_u(t) = &\sum_{i=1}^r P_{ui}\, \tilde{\mu}_i(t) \\ &+ \sum_{m:t_m<t} \sum_{i,j=1}^r P_{ui}\, P_{u_m j}\, A_{u_m u}\, \tilde{g}_{ji}(t - t_m). \end{aligned} \qquad (3)$$

Specifically, if the projection of event type u along the dimension i is given by P_{ui}, then essentially the event type u inherits the natural occurrence rate of events of the component $\tilde{\mu}_i$ with multiplicative weight P_{ui}, that is $\sum_{i=1}^r P_{ui} \tilde{\mu}_i$. In addition, if the projection of event type v along each dimension j is given by P_{vj}, then v's effect on event type u can be evaluated by $\sum_{i,j=1}^r P_{ui} P_{vj} \tilde{g}_{ji}$.

Provided that $r \ll d$, the proposed SLRHP is a simple and straightforward way to: i) impose regularity to the inferred occurrence rates using constraints to the parameters, and ii) reduce the number of parameters. Specifically, the d natural rates and d^2 triggering kernels are reduced to r and r^2, respectively, with the only additional need of inferring the $d \times r$ elements of the matrix P.

Remark on the generality and uniqueness of the projection. Although a projection of the form $g = P \tilde{g} Q$, with

$P \neq Q$ two matrices, is more general than $g = P\tilde{g}P^\top$, it turns out that the latter class is not much smaller than the former. Indeed, any given decomposition $P\tilde{g}Q$ of rank r can be written as a decomposition of rank $2r$ in the form $P'\tilde{g}'P'^\top$, where $P' = (P\ Q)$ and $\tilde{g}' = \left(\begin{smallmatrix} 0 & \tilde{g} \\ 0 & 0 \end{smallmatrix}\right)$ are respectively of size $d \times 2r$ and $2r \times 2r$. Thus, using one matrix P improves the readability and interpretation of the projection, while not leading to substantial differences in the performance of the algorithm. We also remark that, unless any further assumption is made on the projection matrix P or the low-dimensional kernel $\tilde{g}$, the low-rank decomposition of the triggering kernel $g = P\tilde{g}P^\top$ is not unique. More specifically, any change of basis in the r-dimensional space will not alter the decomposition. Notwithstanding, *uniqueness* is not required in order to perform the prediction task, and therefore we do not address this issue in the present paper.

Log-likelihood

General formulation. For $h = 1, ..., H$, let $\mathcal{H}^h = (t_m^h, u_m^h)_{m \leq n_h}$ be the observed i.i.d. realizations sampled from the Hawkes process, and $\mathcal{H} = (\mathcal{H}^h)_{h \leq H}$ the recorded history of events of all realizations. For each realization h, we denote as $[T_-^h, T_+^h]$ the observation period, and u_m^h and t_m^h are respectively the event type and time of occurrence of the m-th event. The log-likelihood of the observations is:

$$\mathcal{L}(P, \mathcal{H}) = \sum_{h=1}^{H} \left[\sum_{m=1}^{n_h} \ln \lambda_{u_m^h}(t_m^h) + \sum_u \int_{T_-^h}^{T_+^h} \lambda_u(s) ds \right]. \quad (4)$$

Our objective is to infer the natural rates $\tilde{\mu}_i$ and triggering kernels $\tilde{g}_{ji}$ of Eq. 3 by means of log-likelihood maximization. From Eq. 3 and Eq. 4, we see that, for arbitrary $\tilde{g}_{ji}$, a single log-likelihood computation already necessitates $O(\sum_{h=1}^{H} n_h^2)$ triggering kernel evaluations. This is intractable when individual realizations can have a number of events of the order 10^7 or 10^8 (e.g. a viral video when modeling information cascades). This issue can be tackled by relying on a convenient K-approximation introduced in [14]. Each natural occurrence rate and kernel function are approximated by a sum of K exponential triggering functions with $\gamma, \delta > 0$ fixed hyperparameter values:

$$\begin{aligned} \widehat{\mu}_i^K(t) &= \textstyle\sum_{k=0}^{K} \beta_{i,k}\, e^{-k\gamma t}; \\ \widehat{g}_{ji}^K(t) &= \textstyle\sum_{k=1}^{K} \alpha_{ji,k}\, e^{-k\delta t}, \end{aligned} \quad (5)$$

Due to the *memoryless property* of exponential functions, this approximation allows for log-likelihood computations with complexity linear in the number of events, i.e. $O(n = \sum_{h=1}^{H} n_h)$. Results of polynomial approximation theory also ensures fast convergence, with respect to K, of the optimal $\widehat{\mu}_i^K$ and $\widehat{g}_{ji}^K$ towards the *true* $\tilde{\mu}_i$ and $\tilde{g}_{ji}$. For instance, if $\tilde{g}_{ji}$ is analytic, then we have for the approximation error: $\sup_{t \in [0,T]} |\widehat{g}_{ji}^K(t) - \tilde{g}_{ji}(t)| = O(e^{-K})$ which means that, for smooth enough functions, setting $K = 10$ already provides a good approximation.

We therefore search the values of parameters α, β that maximize the approximated log-likelihood as well as the most probable projection matrix P, conditionally to the realizations of the process, and under the constraint that the approximated natural rates and triggering kernels remain non-negative. At high-level, this is formally expressed as:

$$\begin{aligned} &\underset{(P,\alpha,\beta)}{\arg\max}\ \widehat{\mathcal{L}}(P, \mathcal{H}; \alpha, \beta) \\ &\text{s.t. } \forall i, j, t: \ \widehat{\mu}_i^K(t) \geq 0 \ \text{ and } \ \widehat{g}_{ji}^K(t) \geq 0. \end{aligned} \quad (6)$$

Above, for clarity of notation, we actually reformulate the log-likelihood by introducing $\widehat{\mathcal{L}}$ that makes implicit the dependency of $\mathcal{L}$ in the fixed hyperparameters K, δ, and γ of Eq. 5. Note also that limiting K and r to small values can be seen as a form of regularization, although more refined approaches could be considered in case of training with datasets of very limited size.

Simplification with tensor notation. In order to perform inference efficiently, we now reformulate the log-likelihood using very large and sparse tensors. We also introduce the artificial $(r+1)$-th dimension to the embedding space in order to remove linear terms of the equation and store the β parameters as additional dimensions of α. In detail, let $\alpha_{(r+1)i,k} = \beta_{i,k}$, $\alpha_{j(r+1),k} = 0$, and $P_{(d+1)i} = \mathbb{1}_{\{i=r+1\}}$ (let $\mathbb{1}_{\{\cdot\}}$ denote the indicator function), also, $\forall u \in \{1, ..., d\}$, $P_{u(r+1)} = 0$. The log-likelihood of the model can then be rewritten as follows:

$$\begin{aligned} \widehat{\mathcal{L}}(P, \mathcal{H}; \alpha) = &\sum_{h,m} \ln \left(\sum_{u,v,i,j,k} P_{ui}\, P_{vj}\, \alpha_{ji,k}\, D_{h,m,u,v,k} \right) \\ &- \sum_{h,u,v,i,j,k} P_{ui}\, P_{vj}\, \alpha_{ji,k}\, B_{h,u,v,k}, \end{aligned} \quad (7)$$

where

$$B_{h,u,v,k} = \begin{cases} \sum_{m=1}^{n_h} J_{v,u,m} f_{k\delta}(T_+^h - t_m^h) & \text{if } v \leq d; \\ f_{k\gamma}(T_+^h - T_-^h) & \text{if } v = d+1; \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

$$D_{h,m,u,v,k} = \begin{cases} \sum_{l=1}^{n_h} I_{h,m,l,u,v}\, e^{-k\delta(t_m^h - t_l^h)} & \text{if } v \leq d; \\ \mathbb{1}_{\{u_m^h = u\}} e^{-k\gamma(t_m^h - T_-^h)} & \text{if } v = d+1; \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

with $f_{kx}(t) = \frac{1 - e^{-kxT}}{kx}$, for x in $\{\gamma, \delta\}$;
$J_{v,u,m} = \mathbb{1}_{\{v = u_m^h\}} A_{vu}$;
$I_{h,m,l,u,v} = \mathbb{1}_{\{u = u_m^h \ \wedge\ v = u_l^h \ \wedge\ t_l^h < t_m^h\}} A_{vu}$.

What the expression suggests is the possibility to optimize the approximated log-likelihood, according to the different parameters and projection matrices, by first creating two large and sparse tensors B and D with four and five dimensions, respectively.

The inference algorithm

The inference is performed by alternating optimization between the projection matrix P and Hawkes parameters α. When all other parameters are fixed, the optimization w.r.t. α is performed using self-concordant function optimization

Algorithm 1 SLRHP Inference: high-level description

Input: $\mathcal{H}$, K, γ, δ, P, α
Output: P, α
Compute D and B *// see the Appendix*
for $i = 1$ **to** *num_iters* **do**
  $\alpha = \arg\max_\alpha \widehat{\mathcal{L}}(P, \mathcal{H}; \alpha)$
    s.t. $\widehat{\mu}_i^K \geq 0$ and $\widehat{g}_{ji}^K \geq 0,\ \forall i, j = 1, ..., r$
  $P = \arg\max_P \widehat{\mathcal{L}}(P, \mathcal{H}; \alpha)$
end for
return P, α

with self-concordant barriers. The technical difficulty of this part is due to the need to ensure that non-negativity constraints are respected. For the optimization w.r.t. P, we introduce novel optimization techniques based on a minorize-maximization algorithm. Alg. 1 outlines the general scheme of our algorithm and details are provided in the Appendix.

Computing B, D tensors. For the inference algorithm to be tractable, special attention has to be paid to the computation of the sparse tensors $B = (B_{h,u,v,k})$ and $D = (D_{h,m,u,v,k})$. Our algorithm has complexity $O(nK\Delta)$, where Δ the maximum node degree in $\mathcal{G}$, and is provided in the Appendix. If $\mathcal{G}$ is sparse, as it is usual for social networks for instance, then $\Delta \ll d$ and hence $O(nK\Delta) \ll O(nKd)$. Thus, storing and computing B and D is tractable for large dense graphs and for particularly large sparse graphs. Note that, since computing the log-likelihood requires the update of occurrence rates at each event time, which in turn depends on the occurrences of all preceding events, the linear complexity in the number of events is only possible due to the memoryless property of the decomposition over a basis of exponentials.

Hawkes parameters optimization. Updating the Hawkes parameters α requires solving the problem:

$$\begin{aligned} \alpha = \arg\max_\alpha \sum_{h,m} \ln\left(c^{hm\top}\alpha\right) - b^\top \alpha \\ \text{s.t } \widehat{\mu}_i^K \geq 0 \text{ and } \widehat{g}_{ji}^K \geq 0,\ \forall i, j = 1, ..., r \end{aligned} \tag{10}$$

where $c_{ijk}^{hm} = \sum_{u,v} P_{ui} P_{vj} D_{h,m,u,v,k}$;
$b_{ijk} = \sum_{u,v,h} P_{ui} P_{vj} B_{h,u,v,k}$.

For the sake of inference tractability we relax the non-negativity constraint and only impose it for the observed time differences:

$$\sum_{k=1}^{K} \alpha_{ji,k} D_{h,m,u,v,k} \ \geq\ 0. \tag{11}$$

Then, we approximate the constrained maximization problem by an unconstrained one, using the concept of *self-concordant barriers* [19]: i.e. we choose $\epsilon > 0$ and solve:

$$\alpha = \arg\max_\alpha \sum_{h,m} \left(\ln\left(c^{hm\top}\alpha\right) + \epsilon b(\alpha)\right) - b^\top \alpha, \tag{12}$$

$$\text{where} \quad b(\alpha)_{hm} = \sum_{i,j,u,v} \ln\left(\sum_{k=1}^{K} \alpha_{ji,k} D_{h,m,u,v,k}\right). \tag{13}$$

A feature of the optimization problem in Eq. 12 is that it verifies the *self-concordance property*. Self-concordant functions have the advantage of behaving nicely with barrier optimization methods and are among the rare classes of functions for which explicit convergence rates of Newton methods are known [20]. This is the reason why we chose to perform the unconstrained optimization using Newton's method, which requires $O(nKr^2 + K^3r^6)$ operations. Note that, since we have n events and aim to learn K Hawkes parameters per pair of groups, we have necessarily $Kr^2 \ll n$. For the second factor of the expression, if we do not have $K^2r^4 \ll n$, we can then reduce the complexity by using quasi-Newton methods that necessitates only $O(nKr^2 + K^2r^4) = O(nKr^2)$ operations. The computation of c, b and $b(\alpha)$ requires multiplying sparse matrices of $O(nK\Delta)$ non-zero elements with a full matrix of r columns, which yields a $O(nK\Delta r)$ complexity. Overall, the complexity of the Hawkes parameters optimization is of the order $O(nKr(\Delta + r))$.

Projection matrix optimization. Let p be a reshaping of the projection matrix P to a vector (linearized). Then, p is updated by solving the following maximization procedure:

$$p = \arg\max_p \sum_{h,m} \ln\left(p^\top \Xi^{hm} p\right) - p^\top \Psi p, \tag{14}$$

where $2\,\Xi_{ui,vj}^{hm} = \sum_k (\alpha_{ji,k} D_{h,m,u,v,k} + \alpha_{ij,k} D_{h,m,v,u,k})$;
$2\,\Psi_{ui,vj} = \sum_{h,k} (\alpha_{ji,k} B_{h,u,v,k} + \alpha_{ij,k} B_{h,v,u,k})$.

The maximization task is performed by a novel minorize-maximization procedure which is summarized by the following proposition and is proved in the Appendix.

Proposition 1. *The log-likelihood is non-decreasing under the update:*

$$p_{ui}^{t+1} = p_{ui}^t \left(\sum_{h,m} \frac{(\Xi^{hm} p^t)_{ui}}{p^{t\top} \Xi^{hm} p^t (\Psi p^t)_{ui}}\right)^{1/2}. \tag{15}$$

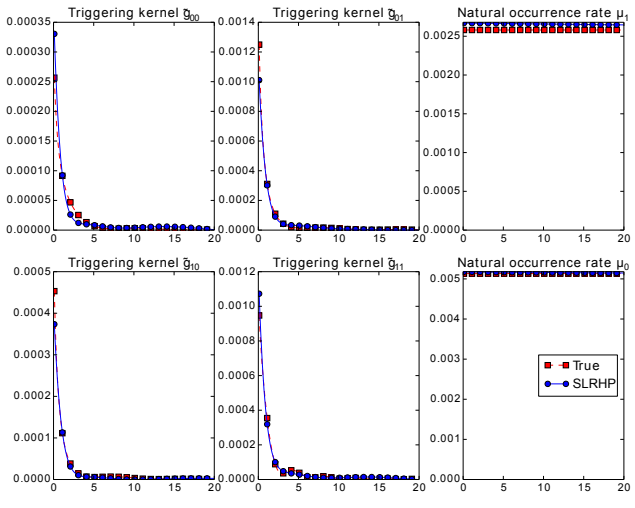
Furthermore, if p_{ui} is a stable fixed point of Eq. 15, then p_{ui} is a local maximum of the log-likelihood.

As previously, computing Ξ, Ψ, and all the matrix-vector products, requires $O(nK\Delta r^2)$ operations, and each update necessitates $O(nd)$ operations. As we consider scenarios where there are at least a few events per dimension, the total complexity of the group affinities optimization is $O(nK\Delta r^2)$. In total, the complexity of our optimization procedure is of the order $O(nK\sigma + nK\Delta r^2)$ and its behavior is linear w.r.t. the number of events and the number of dimensions.

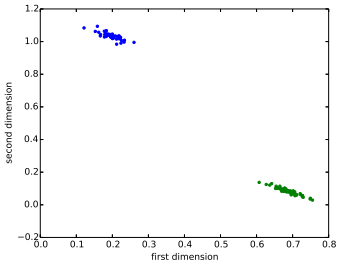
Experiments

Synthetic data

In this section we illustrate the validity and precision of our method in learning the diffusion parameters of simulated Hawkes processes. More specifically, we simulate MHPs such that event types are separated into two groups of similar activation pattern. In the context of social networks, these groups may encode *influencer-influencee* types of relations. We show that our inference algorithm can recover the groups and the corresponding triggering kernels consistently and



(a)



(b)

Figure 1. (a) True and inferred triggering kernels $\tilde{g}_{ij}$ and natural occurrence rates $\tilde{\mu}_i$, for the synthetic dataset. (b) Low-dimensional embedding of the event types learned by SLRHP in the synthetic dataset. The two groups (blue and green) of event types are successfully identified.

with high accuracy. Note that SLRHP is more generic than this specific setting we deploy, and this comes from the fact that there are many notions of '*data structure*' that can be captured by the low-rank approximation. However, we believe that the reported scenario is simple and intuitive, and may therefore provide a clear overview of the capabilities of our approach.

Data generation procedure. The employed procedure for generation of synthetic datasets follows. We assume that the MHPs take place on a random Erdös-Rényi [21] network of $d = 100$ event types whose adjacency matrix A is generated with parameter $p = 0.1$ (i.e. 10 neighbors in average). Then, we consider two distinct groups of event types, and assign each event type to one of the groups at random. The natural occurrence rate $\tilde{\mu}_i$ of each group is fixed to a constant value chosen uniformly over $[0, 0.01]$. The triggering kernels between two groups, i and j, are generated as:

$$\tilde{g}_{ij}(t) = \nu_{ij} \frac{\sin\left(\frac{2\pi t}{\omega_{ij}} + \frac{\pi}{2}((i+j) \bmod 2)\right) + 2}{3(t+1)^2}, \qquad (16)$$

where ω_{ij} and ν_{ij} are sampled uniformly over $[1, 10]$ and $[0, 1/50]$, respectively. These parameter intervals are chosen so that the behavior of the generated process is *non-explosive* [22]. The rationale behind the kernels in Eq. 16 is that they present a power-law decreasing intensity that allows long term influence with a periodic behavior. This kind of dynamics could, for instance, represent the daytime cycles of internet users.

Results. Following the above procedure we generate 8 datasets by sampling 8 different sets of parameters $\{(\omega_{ij}, \nu_{ij})_{i \leq r, j \leq r}, \ (\tilde{\mu}_i)_{i \leq r}\}$. Finally, we simulate 10^5 i.i.d. realizations of the resulting Hawkes process, that we use as training set. The ability of SLRHP to recover the true group triggering kernels $\tilde{g}_{ij}$ is shown in Fig. 1(a) and evaluated by means of the *normalized* L^2 *error*: $\frac{1}{r^2} \sum_{i,j} \frac{||\widehat{g}_{ij} - \tilde{g}_{ij}||_2}{||\widehat{g}_{ij}||_2 + ||\tilde{g}_{ij}||_2}$. On average, this is only 12.9%, with minimum 9.2% and maximum 18.9% amongst the 8 sample datasets. Moreover, the figure compares visually the fitness of the inferred to the true natural occurrence rates and triggering kernel functions.

In order to find the group assignments, we infer the parameters of an SLRHP of rank $r = 2$, and recover the group structure by a clustering algorithm on the projected event types. Then, choosing as basis of the two-dimensional space the centers of the two clusters enables the recovery of the group triggering kernels. Fig. 1(b) shows the 2D embedding learned by our inference algorithm for one of the 8 sample datasets. Two particularly separate clusters appear, which indicates that the group assignments were perfectly recovered. The other 7 datasets gave similar results. These results provide strong indication regarding the validity of our algorithm for inferring the underlying dynamics of MHPs in situations where there is structure in the interactions between the dimensions.

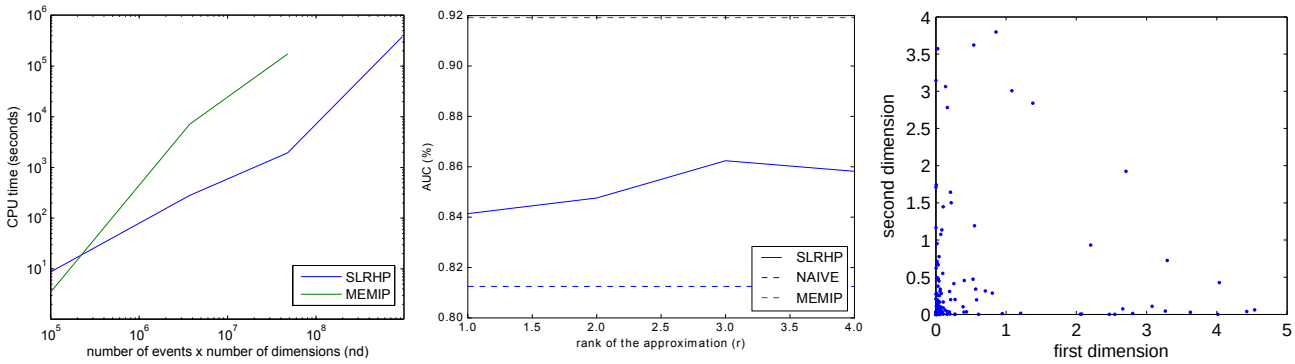
Results on the MemeTracker dataset

Our final set of experiments are conducted on the MemeTracker [23] dataset. MemeTracker is a benchmark corpus of $9.6 \cdot 10^6$ blog posts published between August 2008 and April 2009. We use posts from the period August 2008 to December 2008 as training set, and evaluate our models on the four remaining months. An *event for website* u is defined as the creation of a post on website u containing a hyperlink towards any other website. We also consider that an edge exists between two websites if at least one hyperlink exists between them in the training set. In order to compare the inference algorithms on datasets of different size, prediction was performed on four subsets of the MemeTracker dataset (smaller to larger): MT_1, MT_2, MT_3, and MT_4. These subsets are created by removing the events taking place on websites that appear less than a fixed number of times in the training set. This threshold value (thd in Tab. 1) is, respectively, 50000, 10000, 5000, and 1000.

Prediction task. The task consists in predicting the *next website to create a post*. More specifically, for each event of the test dataset, we are interested in predicting the website on which it will take place knowing its time of occurrence. For MEMIP and SLRHP, the prediction is achieved by scoring the websites according to $\lambda_u(t_m)$, since this value is proportional to the theoretical conditional probability for event m to be of type u. We evaluate the prediction with two met-

Table 1. Experiments on MemeTracker subsets. *AUC* (%) and *Accuracy* (%) are reported for predicting the *next event to happen*, using SLRHP, MEMIP, and NAIVE approach. The experiments denoted with '$*$' did not finish in reasonable time.

	Dataset			***Training Time (secs)***		***AUC***			***Accuracy***		
Name	*thd*	n	d	SLRHP	MEMIP	SLRHP	MEMIP	NAIVE	SLRHP	MEMIP	NAIVE
MT_1	50000	7311	13	8.34	3.16	86.3	86.4	86.1	99.2	99.2	93.1
MT_2	10000	74474	80	281	$7.14 \cdot 10^3$	90.8	92.6	84.4	89.8	92.7	70.6
MT_3	5000	277914	172	$1.95 \cdot 10^3$	$1.74 \cdot 10^5$	86.2	91.9	81.2	87.0	91.6	67.7
MT_4	1000	875402	1075	$3.77 \cdot 10^5$	*	87.0	*	85.2	84.7	*	81.3



(a) (b) (c)

Figure 2. Experimental results on real data: (a) A plot in log-log scale with the training time (secs) for SLRHP and MEMIP algorithm against the quantity nd. The linear behavior for SLRHP and super-linear for MEMIP are clearly visible. (b) Sensitivity analysis of SLRHP accuracy w.r.t. the rank r of the approximation used for inference, and a comparison to the best scores for MMEL and NAIVE baselines on the MT_3 dataset. (c) Low-dimensional embedding of the event types learned by SLRHP for the MT_3 dataset with $r = 2$.

rics: the area under the ROC curve (AUC) and a classification accuracy with a fixed number of candidate types. Due to the high bias towards major news websites (e.g. CNN), the number of candidate types has to be relatively large to see the difference in the performance of algorithms, and we set this value to 30% of the total number of event types d in our experiments. This means that we consider a "*successful prediction*" if the website that eventually fires was ranked by an algorithm in the top 30% candidates.

Baselines. In the following experiments, we use as main competitor the state-of-the-art MEMIP algorithm [14], which is, to the best of our knowledge, the only inference algorithm with linear complexity in the number of events n in the training history. Also, previous work [14] shows that this algorithm outperforms the more standard inference algorithm MMEL [13] on the MemeTracker dataset. In addition, we also use the NAIVE baseline which ranks the nodes according to their frequency of appearance in the training set. Note that this is equivalent to fitting a Poisson process and, hence, does not consider mutual-excitation.

Results. Tab. 1 summarizes the experimental results comparing the proposed SLRHP against MEMIP and NAIVE algorithms on four subsets of the MemeTracker dataset. In each row, the table describes the dataset characteristics, and for each method it provides the training time, AUC, and accuracy with the best parameter settings (for SLRHP, $K = 6$ and $r = 2$, except for MT_3 for which $r = 3$). On small to medium-sized datasets (MT_1, MT_2, MT_3), SLRHP is as efficient as its main competitor MEMIP, while orders of magnitude faster. On the larger MT_4 dataset, SLRHP still runs in reasonable time while outperforming the NAIVE baseline. Note that MEMIP could not be computed in reasonable time for this dataset (less than a few days).

Fig. 2(a) shows in log-log scale the computational time needed for the inference algorithm on all the MemeTracker datasets, with respect to nd. This time is indeed linear in nd for SLRHP, while super-linear for the state-of-the-art competitor of the related literature. In Fig. 2(b) it is indicated that the accuracy measurements are relatively stable with regards to the rank of the approximation r, with a maximum for $r = 3$. Finally, Fig. 2(c) shows the 2D embedding learned by SLRHP for the MT_3 dataset with $r = 2$. In the embedding space, the websites seem to align along the axes of the embedding space, with varying amplitudes. This may indicate that the algorithm recovered two different groups, each one representing similar activities, although with a large variability in the activity of the websites inside each group.

Conclusion

This work focused on modeling multivariate time series where a very large number of event types can occur, and a very large number of historical observations are available for training. The introduced framework is called *Scalable Low-Rank Hawkes Processes* (SLRHP), for which we developed a novel inference algorithm for parameter estimation. Theoretical complexity analysis as well as experimental results show that our approach is highly scalable, while also still competitive with regards to predictive performance as compared to state-of-the-art inference algorithms.

References

[1] David Oakes. The Markovian self-exciting process. *Journal of Applied Probability*, pages 69–77, 1975.

[2] Thomas Josef Liniger. *Multivariate Hawkes processes*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 18403, 2009.

[3] David Vere-Jones. Earthquake prediction – statistician's view. *Journal of Physics of the Earth*, 26(2):129–146, 1978.

[4] Patricia Reynaud-Bouret, Vincent Rivoirard, Franck Grammont, and Christine Tuleau-Malot. Goodness-of-fit tests and nonparametric adaptive estimation for spike train analysis. *Journal of Mathematical Neurosciences*, page 4:3, 2014.

[5] Luc Bauwens and Nikolaus Hautsch. *Modelling financial high frequency data using point processes*. Springer, 2009.

[6] Aurélien Alfonsi and Pierre Blanc. Dynamic optimal execution in a mixed-market-impact hawkes price model. *Finance and Stochastics*, 20(1):183–218, 2015.

[7] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[8] Pierre Brémaud and Laurent Massoulié. Stability of nonlinear Hawkes processes. *The Annals of Probability*, pages 1563–1588, 1996.

[9] Charles Bordenave and Giovanni Luca Torrisi. Large deviations of Poisson cluster processes. *Stochastic Models*, 23(4):593–625, 2007.

[10] Emmanuel Bacry and Jean-François Muzy. Hawkes model for price and trades high-frequency dynamics. *Quantitative Finance*, 14(7):1147–1166, 2014.

[11] Emmanuel Bacry and Jean-François Muzy. First- and second-order statistics characterization of hawkes processes and non-parametric estimation. *IEEE Transactions on Information Theory*, 62(4):2184–2202, 2016.

[12] Niels Richard Hansen, Patricia Reynaud-Bouret, and Vincent Rivoirard. Lasso and probabilistic inequalities for multivariate point processes. *Bernoulli*, 21(1):83–143, 2015.

[13] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional Hawkes processes. In *Proceedings of International Conference on Machine Learning*, ICML, pages 1301–1309, 2013.

[14] Remi Lemonnier and Nicolas Vayatis. Nonparametric markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate Hawkes processes. In *Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer, 2014.

[15] Long Tran, Mehrdad Farajtabar, Le Song, and Hongyuan Zha. Netcodec: Community detection from individual activities. In *Proceedings of the SIAM International Conference on Data Mining*, SIAM ICDM, pages 91–99, 2015.

[16] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems 28*, NIPS, pages 3492–3500, 2015.

[17] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *Proceedings of the International Conference on Machine Learning*, ICML, pages 1717–1726, 2016.

[18] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, AISTATS, pages 641–649, 2013.

[19] Yurii Nesterov, Arkadii Semenovich Nemirovskii, and Yinyu Ye. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.

[20] Stephen Poythress Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[21] P Erdős and A Rényi. On the evolution of random graphs. *Selected Papers of Alfréd Rényi*, 2:482–525, 1976.

[22] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes*. Springer, 2007.

[23] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, 2014.