The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Hawkes Process Inference with Missing Data

Christian R. Shelton
University of California, Riverside
cshelton@cs.ucr.edu

Zhen Qin
University of California, Riverside
tzqin001@cs.ucr.edu

Chandini Shetty
University of California, Riverside
cshet001@cs.ucr.edu

Abstract

A multivariate Hawkes process is a class of marked point processes: A sample consists of a finite set of events of unbounded random size; each event has a real-valued time and a discrete-valued label (mark). It is self-excitatory: Each event causes an increase in the rate of other events (of either the same or a different label) in the (near) future. Prior work has developed methods for parameter estimation from complete samples.

However, just as unobserved variables can increase the modeling power of other probabilistic models, allowing unobserved events can increase the modeling power of point processes. In this paper we develop a method to sample over the posterior distribution of unobserved events in a multivariate Hawkes process. We demonstrate the efficacy of our approach, and its utility in improving predictive power and identifying latent structure in real-world data.

Marked Point Processes

Many applications work with records of events and their times: social networks (information propagation or network changes), computer systems (system calls, hardware failures, network events), global politics (treaties, armed conflicts), as examples. The times are real-valued, and the event types (or labels or marks) are drawn from a finite set.

The general class of marked point processes (MPPs) provide families of distributions over such data. The Poisson process is the simplest and most familiar, but the event sets from any two non-overlapping intervals of time are independent, and thus it does not provide strong modeling power. More complex MPPs include Cox processes and Poisson cluster processes.

In machine learning, a variety of structured MPPs have been developed including Poisson-networks (Rajaram, Graepel, and Herbrich 2005) and piecewise-constant conditional intensity models (PCIMs) (Gunawardana, Meek, and Xu 2011). We concentrate on Hawkes processes (Hawkes 1971). They have been used in earthquake modeling (Marsan and Lengliné 2008), finance (Linderman and Adams 2014; Bacry, Mastromatteo, and Muzy 2015), social networks (Simma and Jordan 2010; Zhou, Zha, and Song 2013; Perry and Wolfe 2013; DuBois, Butts, and Smyth 2013), influence maximization (Du et al. 2013), armed conflicts (Blundell, Heller, and Beck 2012; Mohler 2013; Linderman and Adams 2014), and topic modeling (He et al. 2015; Guo et al. 2015; Du et al. 2015).

Hidden (unobserved) variables are an essential modeling tool. They can simplify a model, be used to find hidden structure, or express the modeler's prior knowledge. Previous uses of Hawkes processes have used complete event data: all event times are observed. This paper expands the modeler's toolkit by allowing partially and fully unobserved events types. This allows hypothesizing about events associated unobserved actors, communication paths, servers, or other variables. It also allows the full use of datasets in which actors enroll or dropout at various times.

For example, we can add unobserved (but modeled) actors in a social network event process. Or, we can allow for periods of non-observation of known actors (for instance, prior to enrolling in a study). Unobserved events in other arenas might correspond to weather, news, or political events. More generally, a Hawkes process between two event-types, A and B, allows for each to self-excite or to excite the other event-type, producing clusters of events that stem from a single seed event of one type. However, if we add a third type, C, and postulate that the observed times of events for A and B result from the marginalization of a Hawkes process over all three event types, this provides a richer model of the relationship between events of types A and B, which includes relationships where A and B do not trigger each other, but both are temporally correlated (through type C).

We consider the situation in which the set of possible labels (types of events) is known. Yet, for some labels for some periods of time, it is known that any events during that time were unobserved (and during other times, all occurring events for this label were observed). This allows for completely hidden event types, as well as masking certain types during certain time intervals.

While other applications placed priors on the parameters of the Hawkes process or other parts of the model and used Gibbs sampling (or similar) over these parameters (Linderman and Adams 2014; Rasmussen 2013), no work to date has considered unobserved events to the extent that we do in this paper. Yang and Zha (2013) used mixtures of Hawkes processes and developed a variational inference method. However, all events were observed; only their assignments to

mixture components were not. Xu, Luo, and Zha (2017) considered the situation in which all events prior to a specific time are unobserved (left censoring), which is a very specific case of the missingness patterns we allow. Finally, if there are no observed events of any labels and the process has an exponential kernel, a closed-form solution exists (Du et al. 2013). However, no general closed-form solution is known for the evidence patterns we consider here.

Contributions

A likelihood-weighted sampler is natural and straightforward. However, as we demonstrate in our results, it does not perform well in complex tasks. Therefore, we develop a Markov chain Monte Carlo (MCMC) method using auxiliary variables. The auxiliary variables are not only the hidden "chain" of events, which have been considered previously in Hawkes processes, but also additional "thinned" events, adapted from previous samplers for other classes of MPPs (Rao and Teh 2011; 2013; Qin and Shelton 2015). This novel combination is an efficient sampler that performs well in all tested scenarios.

We show the advantage of being able to hypothesize unseen events on real-world data. In particular, we demonstrate that, on gang homicides in Chicago, the addition of hidden labels improves prediction accuracy and reveals structure in the data that cannot be extracted without hidden events.

Multivariate Hawkes Process Background

We assume that the process begins at time $t_0 = 0$ and ends at time T. A (complete) sample from the process can be represented as $x = \{(t_1, l_1), (t_2, l_2), \ldots, (t_n, l_n)\}$ where there are n events (a random quantity) and, for notational convenience, we will let $t_{n+1} = T$ (although there is no event at T). $t_{i-1} < t_i, \forall i$ and thus t_i is the time of the ith event, and l_i is the label (mark) of the ith event. Without loss of generality, we assume that the labels are drawn from the set of integers $\{1, 2, \ldots, L\}$. We let $h_t = \{(t_i, l_i) \in x \mid t_i < t\}$ or the set of all events and labels prior to time t, analogous to the natural filtration for the process. For notational convenience, we let $\mathcal{I}_t = \{i \mid t_i < t\}$, or the set of event indices that occurred before t (also the set of event indices in h_t).

A general discrete-label MPP can be specified through the intensity function $\lambda_l(t, h_t)$ which is the rate of an event of label l at time t and is a function of the absolute time, as well as the history of events up to time t: $\lambda_l(t, h_t) = \lim_{\delta t \to 0} \Pr(\text{event of label } l \text{ in } [t, t + \delta t) \mid t, h_t)/\delta t$. The probability density of sample x is

$$p(x) = \exp\left(-\sum_l \int_0^T \lambda_l(s, h_s)\, ds\right) \prod_{i=1}^{n} \lambda_{l_i}(t_i, h_{t_i}) \ . \tag{1}$$

A (linear) multivariate Hawkes process specifies $\lambda_l(t, h_t) = \mu_l + \sum_{i \in \mathcal{I}_t} \phi_{l_i,l}(t - t_i)$ where μ_l is the base rate of events of label l and $\phi_{l,l'}(t)$, the kernel or transfer function, is the increase in the rate of label l' due to an event of label l t time units ago. We can simplify this to

$$\lambda_l(t, h_t) = \sum_{i \in \mathcal{I}_t^0} \phi_{l_i,l}(t - t_i) \tag{2}$$

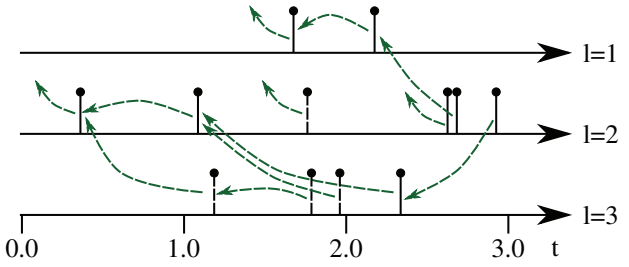


Figure 1: A sample from a multivariate Hawkes process, in black. The green dashed arrows show one possible sampling tree (the arrows point to the parents). Arrows pointing to nowhere indicate that the event was generated by the base rate (root event, $l = 0$).

if we add a special event at t_0 (a time not previously associated with an event). In particular, let $l_0 = 0$ (a new special label), let $\mathcal{I}_t^0 = \mathcal{I}_t \cup \{0\}$, and set the kernel for this new label: $\phi_{l,0}(t) = 0, \forall l$ and $\phi_{0,l}(t) = \mu_l, \forall l > 0$. This event "causes" the base rate of events for each label. For notational compactness, we will assume the kernel has been so redefined.

If we denote $\Phi_{l,l'}(t) = \int_0^t \phi_{l,l'}(s)\, ds$ and $\Phi_{l,\star}(t) = \sum_{l'} \Phi_{l,l'}(t)$, Equation 1 for a Hawkes process is

$$p(x) = \exp\left(-\sum_i \Phi_{l_i,\star}(T - t_i)\right) \prod_i \sum_{j \in \mathcal{I}_{t_i}^0} \phi_{l_j,l_i}(t_i - t_j). \tag{3}$$

Kernels

Multivariate Hawkes processes are usually defined in terms of a base kernel, $\phi(t)$, and an $L \times L$ matrix of non-negative values, M: $\phi_{l,l'}(t) = M_{l,l'}\phi(t)$. In systems with a large number of labels, M is usually sparse: Each event label only excites a small subset of other event labels. The two most common base kernels are the exponential (with parameter $\beta > 0$): $\phi(t) = e^{-\beta t}$ and the power-law (with parameters $\beta > 0$ and $\gamma > 0$): $\phi(t) = (t + \gamma)^{-(1+\beta)}$. The process is well behaved (with probability 1 there are a finite number of events on any finite interval of time) if $\lambda \int_0^\infty \phi(s)\, ds < 1$, where λ is the largest eigenvalue of M (Bacry, Mastromatteo, and Muzy 2015).

If the base kernel is an exponential, the resulting process can be viewed as a Markovian process over a continuous vector-valued state space (essentially tracking the sum of $\phi_{l,l'}(t)$ over all previous events), see Oakes (1975) and Proposition 2 of Bacry, Mastromatteo, and Muzy (2015) for more details. This can reduce the running time of sampling or likelihood sampling (but not our MCMC sampler). We used this improved method for exponential kernels in our experimental results, but otherwise the details are not relevant and are left for the supplementary material.

Unconditional Sampling

The Poisson superposition principle states that the union of events from two independent Poisson processes is itself a Poisson process whose rate function is the sum of the

rate functions of the two underlying processes. Equation 2 shows the rate as the sum of independent rates (one for each past event). Each event generates a set of *children* events independently (at time t the rate of an event is the sum of the rates of any previous event generating a child at time t). This view of a Hawkes process is well established (Hawkes and Oakes 1974) and critical to the development of our sampler.

The sampling algorithm can therefore be recursive in nature. We start with the special label 0 at time 0 and proceed until time T, sampling "children" events from the base rates. Each of these children recursively generates its own events from the kernel rate function. An event (t, l) generates a set of events independently for each other label l' from the inhomogeneous Poisson process with rate $\lambda(t') = M_{l,l'}\phi(t' - t), \forall t' > t$.

This recursive structure forms a tree of events; each event has a parent whose kernel rate function was used to generate it. Figure 1 show one possible sample, along with the (normally discarded) information about which events recursively generated which other events in dashed green.

Posterior Sampling

Our goal is to reason about the distribution of such a process, conditioned on observations of the events for only some labels over some intervals of time. That is, we assume that we observe all elements of x that fall within certain observed label-time ranges.

We let the observational evidence, $z = (z^{(x)}, z^{(o)})$, be such a partial sample which specifies the events during certain time intervals for certain labels. $z^{(x)} = \{(t_1, l_1), \ldots, (t_m, l_m)\}$ is the set of observed events ($z^{(x)} \subseteq x$), analogous to x, but specifying $m \leq n$ events. $z^{(o)} = \{(r_1^s, r_1^e, l_1), \ldots, (r_k^s, r_k^e, l_k)\}$ specifies k observed intervals. In particular, the ith element of $z^{(o)}$ specifies that all events of label l_i on $t \in [r_i^s, r_i^e)$ were observed. We assume the data are missing at random: $z^{(o)} \perp x$. We let $z_l^{(o)}$ denote the union of the intervals over which label l is observed.

Our goal is to estimate $p(x \mid z)$. Note that while the unconditional sampler could sample each set of children independently, conditioned on evidence, these samples are no longer independent. The conditional process $p(x \mid z)$ is not a Hawkes process.

Markov Chain Monte Carlo

A likelihood-weighted sampler is straight-forward (just restrict the sample generation to agree with the evidence), but (as shown later), this method does not perform well on problems of even moderate size.

Auxiliary Variables We use Metropolis-Hastings sampling (Metropolis et al. 1953; Hastings 1970) on the tree representation of the unconditional sampler from above. For an effective MCMC sampler, we introduce two sets of auxiliary variables.

The first auxiliary variable set records the parent structure of the recursive, generational view of a Hawkes process (the green arrows in Figure 1). This set has been used in prior work (Veen and Schoenberg 2008; Marsan and Lengliné 2008) to estimate the parameters of a Hawkes process with *complete data* using expectation-maximization.

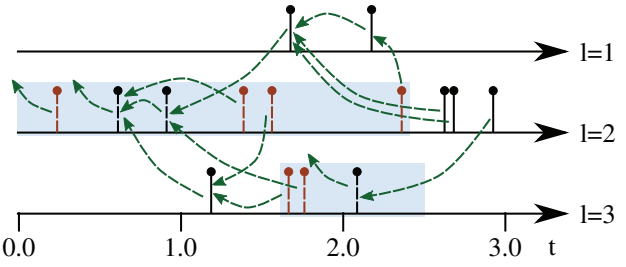


Figure 2: An example sample from the full MCMC process given missing data: Label 2 is unobserved until time 2.4 and label 3 is unobserved from time 1.6 until time 2.5. Black events are x (evidence are solid, sampled are dashed). The parent auxiliary variables, a and $\tilde{a}$, are in green. The virtual events, $\tilde{x}$, are in orange.

The second auxiliary variable set consists of "virtual events," potential (but not realized) times for events, similar to those used for continuous-time Markov processes (Rao and Teh 2011; 2013) and PCIMs (Qin and Shelton 2015). The virtual events are fast to generate (as they are not part of the evidence) and provide a finite number of potential events to turn into a real event (through a sampler move). They are resampled (through other sampler moves) to allow for potentially any real-valued event time. In this way, the times for posterior events that might couple two observed events can be search efficiently without considering the uncountably infinite number of potential event times.

While each type has been used before, they have not previously been used together. The parent auxiliary variables allow for a decoupling of the likelihood (much like mixture-assignment variables in clustering). And, the virtual events are needed to change the dimensionality of the sampling space in a computationally simple fashion. Together, they allow us to tackle Hawkes processes with unobserved events, which no previous sampling method has addressed.

Parent Auxiliary Variables Let $a = \{a_1, \ldots, a_n\}$ where a_i is the index of the parent of event with index i. If we keep track of this information, the unconditional sampler (from above) can be seen as generating samples from $p(x, a)$, whose marginal $p(x)$ is the desired prior distribution (that is, if we throw away the green arrows in Figure 1 we have a sample over labels and times). For notation, let $c_{i,l}$ denote the children of event i that have label l: $\{(t_j, l) \in x \mid a_j = i\}$ and $c_i = \bigcup_l c_{i,l}$. The joint distribution over x and a has only multiplicative terms:

$$p(x, a) = \prod_i \phi_{l_{a_i}, l_i}(t_i - t_{a_i}) \exp\left(-\Phi_{l_i,\star}(T - t_i)\right) \ .$$

It is straight-forward to show that $\sum_a p(x, a) = p(x)$ (see Equation 3).

Virtual Event Auxiliary Variables These virtual events are potential (but not realized) children for each of the real

events (and the special "root event"). They do not generate their own children (real or virtual) and are not part of x. Let $\tilde{x} = \{(\tilde{t}_1, \tilde{l}_1), \ldots, (\tilde{t}_{\tilde{n}}, \tilde{l}_{\tilde{n}})\}$ denote the set of virtual events. Let $\tilde{a}_i$ denote the index of the parent (a real event) of the ith virtual event. Let $\tilde{c}_{i,l}$ and $\tilde{c}_i$, be analogous to $c_{i,l}$ and c_i, but for the ith virtual event.

The unconditional Hawkes process generates events in $c_{i,l}$ from a Poisson process with rate $\phi_{l_i,l}(t-t_i)$. Our distribution including the auxiliary variables generates the events in $\tilde{c}_{i,l}$ from a Poisson process with rate $\kappa \cdot \phi_{l_i,l}(t-t_i)$. Therefore, unconditioned on evidence, by the Poisson superposition principle, the events in $c_{i,l} \cup \tilde{c}_{i,l}$ are drawn from a Poisson process with rate $(\kappa+1)\cdot\phi_{l_i,l}(t-t_i)$. Furthermore, given an event is from this union, it is a virtual event with probability $\kappa/(\kappa+1)$, regardless of its time.

Complete Joint Process Our total auxiliary process is over $x, a, \tilde{x}$, and $\tilde{a}$. Its marginal over x is the same as the original multivariate Hawkes process. Its joint is

$$p(x,a,\tilde{x},\tilde{a}) = \prod_{i=1}^{n} \exp\left[-(\kappa+1)\Phi_{l_i,\star}(T-t_i)\right] \times \prod_{i=1}^{n} \phi_{l_{a_i},l_i}(t_i - t_{a_i}) \prod_{i=1}^{\tilde{n}} \kappa \cdot \phi_{l_{\tilde{a}_i},\tilde{l}_i}(\tilde{t}_i - t_{\tilde{a}_i}) \ . \tag{4}$$

MCMC Sampler Figure 2 shows a sample from this process. We build an MCMC sampler over the dashed items in this figure: events in x that are in the unobserved periods, a and $\tilde{a}$ (the parent sets of all events), and $\tilde{x}$. We constrain $\tilde{x}$ to also only contain events in during the unobserved periods to reduce computation time because virtual events during an observed period can never become real.

Our sampler maintains a state of $x, a, \tilde{x}$, and $\tilde{a}$. There are four types of events: the root event (label 0), evidence events, sampled events, and virtual events. The first three are members of x and the last of $\tilde{x}$. The root and evidence events cannot be changed, but all of the other variables can. At each iteration of the sampler, we pick an event uniformly at random from $x \cup \tilde{x}$. We then select one of the following three "moves" uniformly at random. If the move does not apply to the event, we consider it a self-transition in the Markov chain.

Move 1: Virtual Children This only applies to events from x. Let the event have index i. For this move, we consider replacing the current set $\tilde{c}_i$ with a new set, $\tilde{c}'_i$ drawn from a Poisson process with rate $\kappa \cdot \phi_{l_i,l}(t-t_i)$. This changes the dimension of the distribution (by adding new or removing variables in $\tilde{x}$ and $\tilde{a}$). However, these new variables are sampled without regard to the current state of the process, and therefore the "Jacobian" correction from reversible-jump MCMC is 1 for this case (Green 1995). This is the only type of dimension-changing move (also used as part of the other moves).

Move 2: Virtualness Assume the event is the pair (t, l). If the event is virtual, we propose changing it to be real (moving it from $\tilde{x}$ to x) and sampling a set of virtual children as in Move 1. The likelihood ratio corresponds to moving a term from the second product to the first product in Equation 4, times the probability of sampling these new virtual children. This second part cancels with the proposal likelihood (same as above). Again, we have a correction ratio corresponding to the change in the number of events.

If the event is not virtual, to assure reversibility, we only propose a change if it is neither the root nor an evidence event, and if it has no non-virtual children. In this case, we propose moving it from x to $\tilde{x}$ and removing its virtual children from $\tilde{x}$.

Move 3: Parent This move only applies to non-root events from x. Resampling parents from among x is straight-forward but misses the gains possible by allowing evidence (or sampled events) to "reach back" and suggest possible events earlier in the timeline. Therefore, we allow virtual events to be selected as parents. If one is, it becomes real, and we sample virtual children for this new parent.

While local parent changes are appealing (they can be proposed in constant time), they are difficult to make reversible because they might insert virtual events between the old and new parent, thus rendering the reverse move non-local. Therefore, we sample a new proposed parent from any event earlier in time. Let the event whose parent is to be resampled be (t, l); let $\tilde{h}_t$ be all previous virtual events; and let $H_t = h_t \cup \tilde{h}_t$. We propose $(t'_p, l'_p) \in H_t$ as the new parent with probability proportional to $w(t'_p, l'_p) = \phi_{l,l'_p}(t - t'_p)$. If $(t'_p, l'_p) \in \tilde{x}$, then our proposal includes moving it to x and sampling (as in Move 1) new virtual children for it, $\tilde{c}'$. If the proposed change would leave the old parent with no real children and it is not the root event, then we propose to change the old parent to be virtual (and remove its children) with probability $\kappa/(\kappa+1)$.

The moves were designed, for the most part, to simplify the acceptance ratios. The exact form of the ratios are straightforward, but tedious to derive. We leave them to the supplemental material.

Synthetic Data Experiments

For evaluation, we used an exponential and a power-law kernel. We generated two different sizes of problems, each in an "easy" and "hard" version. We then tested the time-accuracy trade-offs of the base likelihood weighting and our MCMC method on each of these eight combinations.

Because the algorithms are samplers, estimating the expectation of any function of the sample is possible. We chose the number of events for a particular set of labels because this query is simple and directly related to the average *posterior* rate, which is important in kernel parameter estimation.

We chose the exponential kernel with $\beta = 1$ and the power-law kernel with $\beta = 1$ (inverse squared decay) and $\gamma = 1$, so they integrate to the same quantity. The power-law kernel has a heavier tail, however, and so more of its power will

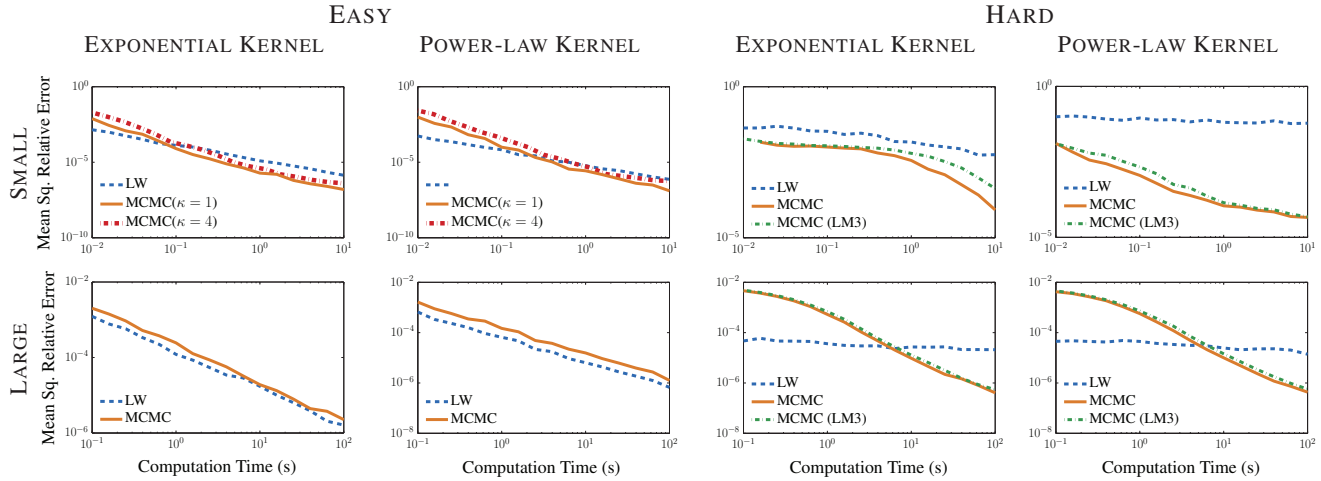


Figure 3: Results on synthetic problems. Both likelihood weighting (LW) and our MCMC sampler perform well on easy problems. LW fails on hard problems, while our MCMC sampler continues to performs well. LM3 is a limited version of "move 3."

fall after time T. We describe the small and large synthetic problems in more detail in the supplementary material. The small problems consist of chains of labels (each affecting the next one or two). The large problems consist of labels connected via a graph with a power-law degree distribution. All problems have evidence on approximately half of the labels.

Methods

Each combination of kernel, problem size, and problem difficulty fixed a particular input process, evidence, and query. We chose 16 geometrically evenly spaced computation times and independently ran the algorithms 200 independent times for each of these computation times, stopping the sampling when the algorithm had reached the computational time limit. For the MCMC method, this computational budget included the burn-in time, which we set to be 1000 iterations for the easy problems and 5000 iterations for hard problems.

Running for a fixed computational time, instead of a fixed number of iterations, can bias the resulting estimator toward samples with smaller numbers of events (because they take less time, generally, to generate). However, our experience did not show this was a factor in these experiments.

For each problem and computational budget, we used the 200 estimates to compute the mean squared relative error (the relative variance). We computed the true values by running the MCMC sampler for 5 hours. Both sampling algorithms can be easily parallelized, but, so as not to complicate the comparison, we ran all experiments on a single core. The single tunable parameter is κ. We set it to 1, following the methodology of similar samplers (Rao and Teh 2011; Qin and Shelton 2015). We illustrate the effect of changing κ on the small-easy problem. We also tried removing the ability of the MCMC "move 3" to select a previously virtual parent and illustrate its effect on the hard problems (limited move 3).

Problem			MCMC	LW	
diff	size	kernel	samp/sec	samp/sec	eff/sec
E	S	exp	3.32×10^6	4.67×10^5	2.02×10^3
E	S	pow	2.84×10^6	5.71×10^5	4.17×10^3
E	L	exp	8.65×10^5	8.53×10^2	1.87×10^2
E	L	pow	7.66×10^5	9.58×10^2	4.34×10^2
H	S	exp	5.61×10^5	1.23×10^4	0.30
H	S	pow	7.14×10^5	1.75×10^4	0.16
H	L	exp	3.51×10^5	1.04×10^2	0.15
H	L	pow	2.12×10^5	1.83×10^1	0.22

Table 1: Generation speed statistics for both algorithms. Column samp/sec lists the average number of samples taken per second (a single step for MCMC and a full sample for LW). Column eff/sec lists the effective number of samples for LW (the average effective sample size across all runs with the longest runtime).

Results

Figure 3 shows the accuracy-time trade-off for the algorithms. On the easy problems (Figure 3, left), both algorithms perform well. The likelihood weighting method has fewer samples and many fewer effective samples, measured as $(\sum_i w_i)^2 / \sum_i w_i^2$, see Table 1. However, the performance suggests this is compensated by the independence of each sample, unlike MCMC. For the small instance, we also show the very small effect of changing κ to 4. We found similarly small differences for adjustments of κ between 0.5 and 8 for all of the experimental designs tested.

On the hard problems (Figure 3, right), things are different. Our MCMC method performs well with expected $O(1/n)$ reduction in variance. The exponential kernel for the small problem requires more than the provided burn-in time (a full burn-in is not always possible in 0.01 seconds, hence the missing point). Thus, the $O(1/n)$ behavior does not start until after 1 second. If we remove the ability of the MCMC sampler to select a previously virtual event as a parent (lim-

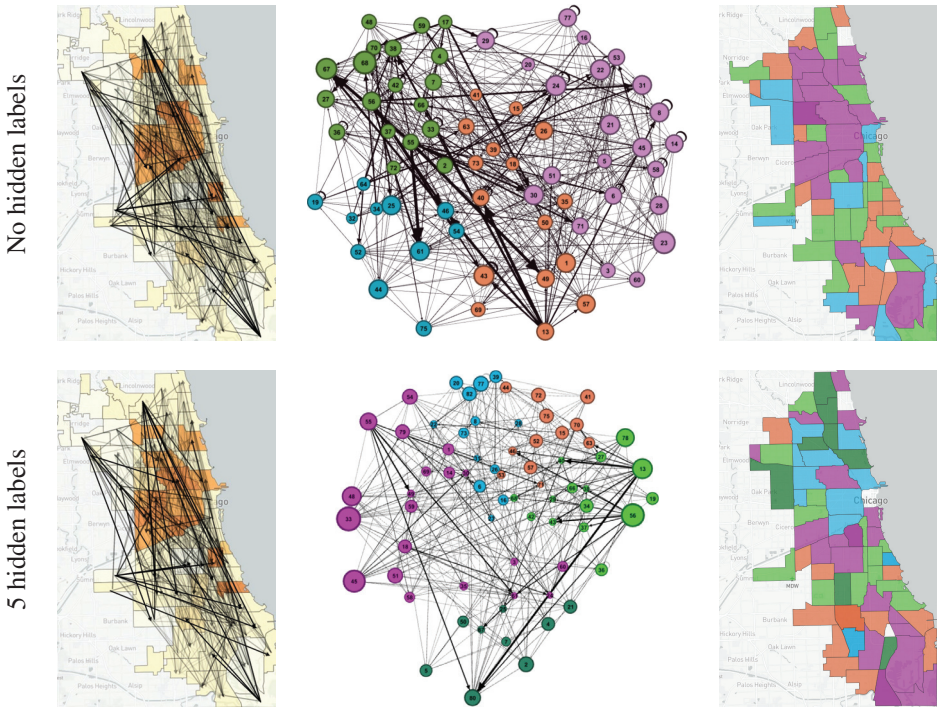


Figure 4: Results for crime data, without hidden event types (top) and with hidden event types (bottom). Background region color on left map shows the (area-normalized) background rates (μ) of the estimated model. Estimated model network (M) shown in three ways: (left) as super-imposed on map with arc darkness proportional to weight in M, (middle) as a filtered graph, laid out to demonstrate automatically found clusters, and (right) the same clusters shown geographically.

ited move 3), the performance degrades when there are more events and the base rate is small, and therefore it does not naturally generate real events. This can be seen in the small hard problem with an exponential kernel. Note that adding this ability never hurts the performance.

The likelihood weighting performs terribly on the hard problems. We do not even see the expected $O(1/n)$ performance. This is because very few samples with any significant weight are generated as seen in Table 1 in the "eff/sec" column. While the bias and variance of this estimator do decrease as $O(1/n)$, 200 runs is not sufficient to demonstrate this effect. Almost never, in all of the 200 runs and all of the samples taken in each of these runs, does the sampler generate a sample that is even remotely likely under the posterior distribution.

All sampling code exploits the sparsity of M.

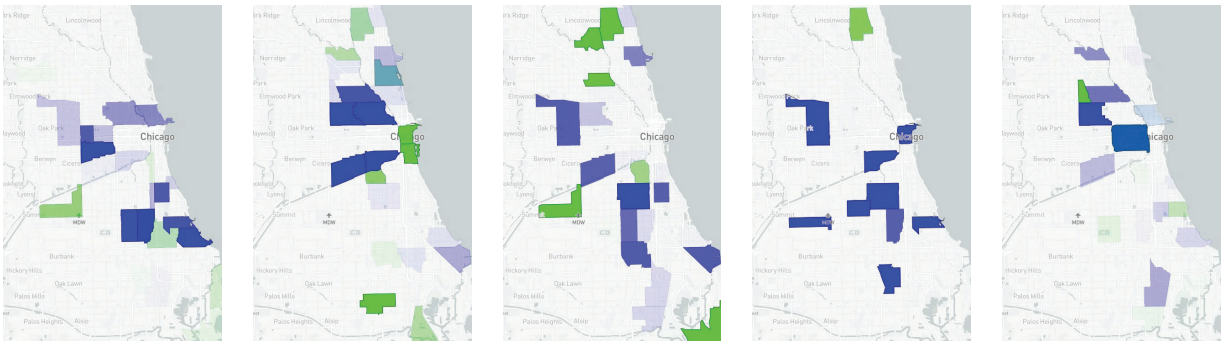
Crime Data Experiments

To demonstrate the utility of unobserved variables, we used homicide data provided by the Chicago Police Department from 1965 through 1995 (Block, Block, and Illinois Criminal Justice Information Authority 2005) filtered to only those events reported to be gang-related. This follows the same methodology as Papachristos (2009). We treat each gang-related homicide as an event with a label corresponding to the community area in which it occurred. There are 77 different Chicago communities identified in the dataset and 2195 events. We did not supply any proximity information about the regions to the model or algorithm.

While a self-excitatory process would seem to be a good fit to gang-related homicides, previous uses of a plain Hawkes process failed to find structure in the data. Others have used a Hawkes process as part of a more complex model to find structure. For instance, Linderman and Adams (2014) placed hierarchical priors over the parameters to produce a clustering of communities. We show that using a plain Hawkes process with unobserved event types (labels) will also find structure, and does better at prediction, relative to a plain Hawkes process.

Methods

We first used a Hawkes process with an exponential kernel to model the 77 fully observed labels (one for each community of Chicago). Because there are no missing data, our sampling method reduces to sampling over the latent generational structure (a). We use this as the expectation step in a Monte Carlo expectation maximization procedure (Wei and Tanner 1990) to estimate the parameters: M (matrix of inter-label weights), β (decay rate of kernel), and μ (vector of base rate for each label). We included an L_1 regularizer on the elements of M with strength set via a search over powers of 10. We let $\kappa = 1$ in our sampler. For the maximization step, given a collection of samples and β, M and μ can be solved in closed form (see supplementary material). We use a 1-dimensional line search to find β. This method is essentially the same as previous EM methods for Hawkes processes with fully observed labels



(a) (b) (c) (d) (e)

Figure 5: The strength of connection to (in green) geographic labels and from (in blue) geographic labels each of the five hidden event labels.

but hidden branching structure (Veen and Schoenberg 2008; Marsan and Lengliné 2008).

To demonstrate the advantage of hidden events, we repeated the same process, but with five extra process labels, each completely unobserved. These can be viewed as missing variables (event sequences) with unspecified semantics. They can be used by the (unaided) learning procedure to simplify the structure, similar to how hidden variables in a Bayesian network can simplify the distribution's representation. Note each hidden event type is not a single scalar random variable, but rather a full (unobserved) sequence of event times.

These processes are allowed arbitrary connections with each other and the 77 observed labels. To encourage their use in the model, we clamped the elements of M corresponding to connections between two observed labels to 0 for the first half of the EM iterations.

Finally, we reran the models, training only on crimes from 1993 and 1994. We then tested on the last year of data (1995), advancing time one event at a time and predicting the location of the next event (the time of the next event is almost always in the next zero to three days and not as interesting to estimate). There are 215 events in 1995 on 171 different days. We credited a model if the region with the highest probability of having the next event corresponded to any of the regions with an event on that day. We used our MCMC method to do the forecasting.

Results

The prediction accuracy for the model with five hidden event types was 22% higher (6.5% versus 5.3%). However, more interesting was the hidden event model's ability to capture sociological structure in the data.

The EM algorithm was very stable, producing very similar results on multiple runs, despite different starting values and random seeds. The resulting β value is $\frac{1}{28.6\ days}$ for the model with no hidden event types and $\frac{1}{48.2\ days}$ for the model with hidden event types. The M and μ values are shown in Figure 4. The result is messy (despite tuning the regularization) and thus consistent with previous attempts. The network among the observed variables is slightly sparser for the model with hidden event types, but it is hard to tell from this figure. We clustered the regions (see supplemental material for method) to try to find meaningful clusters, shown in Figure 4. While some are geographically reasonable, they do not reflect the major connectivity of the model.

Yet, the hidden event connections of the model with 5 hidden labels do directly reveal known gang-related structure. Figure 5 shows the connections in M to and from the five hidden labels. Each plot, therefore, shows regions (in blue) that tend to get "triggered" together (from the same hidden event) and regions (in green) that tend to trigger these hidden events. These results correlate well with previous studies. Block and Block (1993) mapped out Chicago gang crime in the late 1980s. Neither the base rates of models nor clustering of the M matrix (Figure 4) identify the gang areas in south Chicago. However, two of our hidden label connections, Figure 5 (c,d), identify the primary areas in south Chicago also noted in Exhibit 1 of Block and Block (1993). The cluster in Figure 5(b) matches the first cluster (that of highest crime rate) identified by Linderman and Adams (2014). Further, the "Street Gang Turfs" of 1991 highlighted in Exhibit 4 of Block and Block (1993) are also identified in Figure 5(e).

Summary

The code for general inference in Hawkes processes with optional parallelization, as well as the wrapper code to run the exact experiments done here and gather the results are available at https://github.com/cshelton/hawkesinf.

We developed a reversible-jump MCMC sampler for Hawkes processes. It allows the use of hidden event types in Hawkes processes, providing flexibility to modelers. We demonstrated the utility of such hidden events by using them to more accurately predict locations, and to find meaningful clusters of regions in Chicago crime data.

Acknowledgments This work was supported by the Defense Advanced Research Projects Agency (FA8750-14-2-0010) and the National Science Foundation (IIS 1510741).

References

Bacry, E.; Mastromatteo, I.; and Muzy, J.-F. 2015. Hawkes processes in finance. *Market Microstructure and Liquidity*

1(1).

Block, C. R., and Block, R. 1993. Street gang crime in chicago. Research in brief, National Institute of Justice.

Block, C. R.; Block, R. L.; and Illinois Criminal Justice Information Authority. 2005. Homicides in Chicago, 1965–1995. Technical Report ICPSR 6399, Inter-university Consortium for Political and Social Research.

Blundell, C.; Heller, K. A.; and Beck, J. M. 2012. Modelling reciprocating relationships with Hawkes processes. In *NIPS*.

Du, N.; Song, L.; Gomez-Rodriguez, M.; and Zha, H. 2013. Scalable influence estimation in continuous-time diffusion networks. In *NIPS*.

Du, N.; Farajtabar, M.; Ahmed, A.; Smola, A. J.; and Song, L. 2015. Dirichlet-Hawkes processes with applications to clustering continuous-time document streams. In *KDD*.

DuBois, C.; Butts, C. T.; and Smyth, P. 2013. Stochastic blockmodeling of relational event dynamics. In *AI-STATS*.

Green, P. J. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82(4):711–732.

Gunawardana, A.; Meek, C.; and Xu, P. 2011. A model for temporal dependencies in event streams. In *NIPS*.

Guo, F.; Blundell, C.; Wallach, H.; and Heller, K. 2015. The Bayesian echo chamber: Modeling social influence via linguistic accommodation. In *AI-STATS*.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57:97–109.

Hawkes, A. G., and Oakes, D. 1974. A cluster process representation of a self-exciting process. *Journal of Applied Probability* 11(3):493–503.

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

He, X.; Rekatsinas, T.; Foulds, J.; Getoor, L.; and Liu, Y. 2015. HawkesTopic: A joint model for network inference and topic modeling from text-based cascades. In *ICML*.

Linderman, S. W., and Adams, R. P. 2014. Discovering latent network structure in point process data. In *ICML*.

Marsan, D., and Lengliné, O. 2008. Extending earthquakes' reach through cascading. *Science* 319:1076–1079.

Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; and Teller, E. 1953. Equations of state calculations by fast computing machines. *Journal of Chemical Physics* 21(6):1087–1092.

Mohler, G. 2013. Modeling and estimation of multi-source clustering in crime and security data. *The Annals of Applied Statistics* 7(3):1525–1539.

Oakes, D. 1975. The Markovian self-exciting process. *Journal of Applied Probability* 12(1):69–77.

Papachristos, A. V. 2009. Murder by structure: Dominance relations and the social structure of gang homicide. *American Journal of Sociology* 115(1):74–128.

Perry, P. O., and Wolfe, P. J. 2013. Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society: Series B* 75(5):821–849.

Qin, Z., and Shelton, C. R. 2015. Auxiliary Gibbs sampling for inference in piecewise-constant conditional intensity models. In *UAI*.

Rajaram, S.; Graepel, T.; and Herbrich, R. 2005. Poisson networks: A model for structured point processes. In *Proceedings of the AI STATS 2005 Workshop*.

Rao, V., and Teh, Y. W. 2011. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *UAI*.

Rao, V., and Teh, Y. W. 2013. Fast MCMC sampling for Markov jump processes and extensions. *Journal of Machine Learning Research* 14:3207–3232.

Rasmussen, J. G. 2013. Bayesian inference for Hawkes processes. *Methodology and Computing in Applied Probability* 15(3):623–642.

Simma, A., and Jordan, M. 2010. Modeling events with cascades of Poisson processes. In *UAI*.

Veen, A., and Schoenberg, F. P. 2008. Estimation of space-time branching process models in seismology using an EM-type algorithm. *Journal of the American Statistical Association* 103(482):614–624.

Wei, G. C. G., and Tanner, M. A. 1990. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association* 85(411):699–704.

Xu, H.; Luo, D.; and Zha, H. 2017. Learning Hawkes processes from short doubly-censored event sequences. In *ICML*.

Yang, S.-H., and Zha, H. 2013. Mixture of mutually exciting processes for viral diffusion. In *ICML*.

Zhou, K.; Zha, H.; and Song, L. 2013. Learning triggering kernels for multi-dimensional Hawkes processes. In *ICML*.