Hawkes Process Inference with Missing Data: Supplementary Material

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

Zhen Qin
University of California, Riverside
zqin001@cs.ucr.edu

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

Abstract

Supplemental material: state-space formulation, likelihood-weighting sampler, pseudo-code for samplers, derivations of likelihood and acceptance ratios, sample generation speed for experimental results, full synthetic results, derivation of M-step for MCEM, and details of the network cluster analysis.

Background

Equation 3 (main paper) can be derived as

$$
\begin{aligned}
p(x) &= \exp\left(-\sum_l \int_0^T \sum_{j\in\mathcal{I}^0_s} \phi_{l_j,l}(s-t_j)\,ds\right) \\
&\quad\times \prod_i \sum_{j\in\mathcal{I}^0_{t_i}} \phi_{l_j,l_i}(t_i-t_j) \\
&= \exp\left(-\sum_l \sum_i \int_{t_i}^T \phi_{l_i,l}(s-t_i)\,ds\right) \\
&\quad\times \prod_i \sum_{j\in\mathcal{I}^0_{t_i}} \phi_{l_j,l_i}(t_i-t_j) \\
&= \exp\left(-\sum_i \Phi_{l_i,\star}(T-t_i)\right) \\
&\quad\times \prod_i \sum_{j\in\mathcal{I}^0_{t_i}} \phi_{l_j,l_i}(t_i-t_j)
\end{aligned}
$$

where the first step switches the order of the sum and integral by noting that the integral over all time of the sum of all previous events is the same as the sum over all events of the integral over the time after each event.

State Space for Exponential Kernel

In the particular case of a multivariate Hawkes process with an exponential base kernel, the history of events prior to time t can be summarized by a state of L scalars, $s_1(t), \ldots, s_L(t)$. $s_l(t)$ summarizes the contribution to the rate of all past events of label l. In particular, we start $s_l(0) = 0, \forall l$. The system then evolves as $\frac{ds_l(t)}{dt} = -\beta s_l(t)$ when t is not an event time for label l. At event time t_i, $s_{l_i}(t_i^+) = s_{l_i}(t_i^-) + 1$ (that is the state value for l_i is increased by one as we cross the event time). The rate can then be written as $\lambda_{l'}(t, h_t) = \mu_{l'} + \sum_l M_{l,l'} s_l(t)$. See Proposition 2 of Bacry, Mastromatteo, and Muzy (2015) for more details.

Algorithm 1 Unconditional sampling algorithm

```
Sample next event of label $l'$ generated from event of label $l$. Assume parent event is at time 0 and last event was at time $s$. Multiply kernel function by factor $\rho$ ($\rho = 1$ until Algorithm 4).
function SampleNextEvent($l$,$l'$,$s$,$\rho$)
  $r \leftarrow$ SampleFromUnitExp
  return $\bar{\phi}^{-1}_{l,l'}(r/\rho + \Phi_{l,l'}(s))$    ▷ If $\bar{\phi}^{-1}_{l,l'}(s)$ is undefined for $s$, it returns $\infty$.

Sample all descendant events of event at $t_0$ of label $l$ until time $T$
function Sample($l$,$t_0$,$T$)
  $x \leftarrow \{\}$
  for all $l'$
    $t \leftarrow t_0$
    while $t < T$
      $t \leftarrow t_0 +$ SampleNextEvent($l$,$l'$,$t - t_0$,1)
      if $t < T$
        $x \leftarrow x \cup \{(t,l')\} \cup$ Sample($l'$,$t$,$T$) ▷ Add event and all of its descendants.
  return $x$
```

Unconditional Sampling

Algorithm 1 is the unconditional sampler for a Hawkes process. It depends upon the ancestor interpretation of a Hawkes process. In this view, each event of label l at time t independently generates "children" events of label l' from an inhomogeneous Poisson process with rate $\phi_{l,l'}(t'-t)$ at time $t' > t$. This is equivalent to having a net total rate at time t' for label l' as $\sum_{i|t_i<t'} \phi_{l_i,l'}(t'-t_i)$.

As mentioned in Section (main paper), we fold the background rate for label l, μ_l, into these sums by adding an special label $l = 0$ and a single event at the start of the sequence with label 0 (no other event has this label). We call this event the root event. For $l = 0$, the kernel is different: $\phi_{0,l'}(t) = \mu_l$.

The function SAMPLENEXTEVENT samples a next event from a inhomogeneous Poisson process by transforming an exponential duration sample (a sample for a homogeneous Poisson process) through the inverse of the cumulative of the kernel. There are other methods, but this is the most straight-forward and, for the kernels commonly used, works well.

Algorithm 2 Likelihood weighting sampling algorithm

```
  Sample descendant events as with SAMPLE, but skip over observed times in $z^{(o)}$
  function SAMPLEWITHEVID($l$,$t_0$,$T$,$z^{(o)}$,$\rho$)
    $e \leftarrow \{\}$
    for all $l'$
      $t \leftarrow t_0$
      while $t < T$
        $t \leftarrow t_0 +$ SAMPLENEXTEVENT($l$,$l'$,$t - t_0$,$\rho$)
        if $t \in z_{l'}^{(o)}$          ▷ Skip over observed times for label $l'$.
          $t \leftarrow$ next unobserved time for $l'$ after $t$
        else if $t < T$
          $e \leftarrow e \cup \{(t, l')\} \cup$ SAMPLEWITHEVID($l'$,$t$,$T$,$\rho$)
    return $e$

  Compute the sample's weight
  function WEIGHT-SAMPLE($x$,$z$)
    $w \leftarrow 1$
    for all $(t, l) \in x$          ▷ each sampled or observed event
      for all $(r^s, r^e, l') \in z^{(o)}$ such that $r^e > t$          ▷ each observed interval after the event
        $w \leftarrow w \cdot \exp(-\Phi_{l,l'}(r^e - t) + \Phi_{l,l'}(\max(0, r^s - t)))$          ▷ weight for duration of interval
    for all $(t, l) \in z^{(x)}$          ▷ each observed event $w' \leftarrow 0$
      for all $(t', l') \in x$ such that $t' < t$          ▷ each earlier event
        $w' \leftarrow w' + \phi_{l',l}(t - t')$          ▷ additive kernel weight
      $w \leftarrow w \cdot w'$
    return $w$

  Sample events from 0 to $T$ consistent with evidence $z$ and return samples and likelihood weight
  function LW-SAMPLE($z$,$T$)
    $x \leftarrow z^{(x)} \cup$ SAMPLEWITHEVID(0,0,$T$,$z^{(o)}$,1)          ▷ Sample descendants of root events.
    for all $(t, l) \in z^{(x)}$
      $x \leftarrow x \cup$ SAMPLEWITHEVID($l$,$t$,$T$,$z^{(o)}$,1)          ▷ Sample descendants of observed events.
    return ($x$, WEIGHT-SAMPLE($x$,$z$))
```

Likelihood Weighting

The most straight-forward method of producing an estimate of the posterior distribution is to use importance sampling. In particular, we sample from the prior distribution, except we will not allow sampled events to occur during observed intervals, and we add events from $z^{(x)}$ (including any sampled descendant events). We let $q(x \mid z)$ represent the distribution induced by this sampling method. To compensate for the differences in sampling distributions, we will weight the corresponding sample by the ratio of its likelihood under the prior to its likelihood under q. Discarding the constant of proportionality, $p(z)$, and noting that $p(x \mid z) = p(x)$ if x is consistent with z (because x contains all of the information in z), $w(x) \propto \frac{p(x)}{q(x|z)}$. $p(x)$ is as in Equation 3 (main paper). $q(x \mid z)$ is much the same, except that we do not integrate $\phi_{l_i,l}(s - t_i)$ over intervals of observed evidence of label l (because we are not sampling during those times for label l) and we do not multiply the second part over i indices which are part of the evidence (because those samples are forced). Therefore, the ratio becomes exactly those components that are excluded from q:

$$
\begin{aligned}
w(x) &\propto \frac{p(x)}{q(x \mid z)} \\
&= \frac{\exp\left(-\sum_l \sum_i \int_{t_i}^{T} \phi_{l_i,l}(s - t_i)\, ds\right)}{\exp\left(-\sum_l \sum_i \int_{t_i}^{T} I[s \notin z_l^{(o)}] \phi_{l_i,l}(s - t_i)\, ds\right)} \\
&\quad \times \frac{\prod_{i | x_i \notin z_x} \sum_{j \in \mathcal{I}_{t_i}^0} \phi_{l_j,l_i}(t_i - t_j)}{\prod_i \sum_{j \in \mathcal{I}_{t_i}^0} \phi_{l_j,l_i}(t_i - t_j)} \\
&= \exp\left(-\sum_{i,l} \int_{s \in [t_i, T) \cap z_l^{(o)}} \phi_{l_i,l}(s - t_i)\, ds\right) \\
&\quad \times \prod_{(t,l) \in z^{(x)}} \sum_{j \in \mathcal{I}_t^0} \phi_{l_j,l}(t - t_j) \ .
\end{aligned}
$$

Algorithm 2 is a direct modification of Algorithm 1 to add likelihood weights. In this version, the weights are calculated after the sample is generated for easy of exposition. It is simple to calculate the weight as the sample is generated.

MCMC

The acceptance ratios can be calculated as follows. Each is the the ratio of the probability of selecting the reverse move to the probability of selecting the forward move (which we will denote $\mathcal{Q}$) multiplied by the ratio of the likelihood of the new state to the likelihood of the old state (which we will denote $\mathcal{L}$).

To ease notation, we let $a_t(t, l)$ and $a_l(t, l)$ be the time and label (respectively) of the parent of event at time t with label l. We let $a'_t(t, l)$ and $a'_l(t, l)$ be the same after the proposed move. For most (t, l) the parents are the same before and after.

Move 1: Virtual Children This move resamples the children for real event (t_i, l_i). The old state had $n + \tilde{n}$ events. The new state samples and adds $|\tilde{c}'_i|$ new virtual events, and drops $|\tilde{c}_i|$ old virtual events. The ratio of the proposal probabilities is the ratio of the chance of selecting this move for event i times the ratio of the probabilities of selecting the (new or old) set of virtual events:

$$\mathcal{Q}_1 = \frac{\frac{1}{3}\frac{1}{n+\tilde{n}+|\tilde{c}_i'|-|\tilde{c}_i|} \times \exp\left(-\kappa \cdot \Phi_{l_i,\star}(T-t_i)\right)}{\frac{1}{3}\frac{1}{n+\tilde{n}} \times \exp\left(-\kappa \cdot \Phi_{l_i,\star}(T-t_i)\right)} \times \frac{\prod\limits_{(t,l)\in\tilde{c}_i} \kappa \cdot \phi_{l_i,l}(t-t_i)}{\prod\limits_{(t,l)\in\tilde{c}_i'} \kappa \cdot \phi_{l_i,l}(t-t_i)}$$

$$= \frac{(n+\tilde{n}) \prod\limits_{(t,l)\in\tilde{c}_i} \kappa \cdot \phi_{l_i,l}(t-t_i)}{(n+\tilde{n}+|\tilde{c}_i'|-|\tilde{c}_i|) \prod\limits_{(t,l)\in\tilde{c}_i'} \kappa \cdot \phi_{l_i,l}(t-t_i)}$$

The likelihood ratio is

$$\mathcal{L}_1 = \frac{\prod\limits_{(t,l)\in x} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right) \phi_{a_l'(t,l),l}(t-a_t'(t,l))}{\prod\limits_{(t,l)\in x} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right) \phi_{a_l(t,l),l}(t-a_t(t,l))} \times \frac{\prod\limits_{(t,l)\in\tilde{x}\cup\tilde{c}_i'\setminus\tilde{c}_i} \kappa \cdot \phi_{a_l'(t,l),l}(t-a_t'(t,l))}{\prod\limits_{(t,l)\in\tilde{x}} \kappa \cdot \phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$= \frac{\prod\limits_{(t,l)\in\tilde{c}_i'} \kappa \cdot \phi_{a_l'(t,l),l}(t-a_t'(t,l))}{\prod\limits_{(t,l)\in\tilde{c}_i} \kappa \cdot \phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$= \frac{\prod\limits_{(t,l)\in\tilde{c}_i'} \kappa \cdot \phi_{l_i,l}(t-t_i)}{\prod\limits_{(t,l)\in\tilde{c}_i} \kappa \cdot \phi_{l_i,l}(t-t_i)}$$

where the last line holds because all of the events in the products are children of event i.

The ratio in $\mathcal{L}_1$ cancels with the similar terms in $\mathcal{Q}_1$ leaving

$$\mathcal{A}_1 = \mathcal{Q}_1\mathcal{L}_1 = \frac{n+\tilde{n}}{n+\tilde{n}+|\tilde{c}_i'|-|\tilde{c}_i|} \;.$$

Move 2: Virtualness We first consider changing an event at time $\tilde{t}$ with label $\tilde{l}$ from virtual to real. This adds $|\tilde{c}'|$ new virtual events (as children from the newly real event). The reverse move removes these same events, but does not add any. The derivations are similar to those of Move 1.

$$\mathcal{Q}_2^{\text{virt}\to\text{real}} = \frac{\frac{1}{n+\tilde{n}+|\tilde{c}'|}}{\frac{1}{n+\tilde{n}} \exp\left(-\kappa \cdot \Phi_{\tilde{l},\star}(T-\tilde{t})\right) \prod\limits_{(t,l)\in\tilde{c}'} \kappa \cdot \phi_{\tilde{l},l}(t-\tilde{t})}$$

$$= \frac{n+\tilde{n}}{(n+\tilde{n}+|\tilde{c}'|) \exp\left(-\kappa \cdot \Phi_{\tilde{l},\star}(T-\tilde{t})\right) \prod\limits_{(t,l)\in\tilde{c}'} \kappa \cdot \phi_{\tilde{l},l}(t-\tilde{t})}$$

$$\mathcal{L}_2^{\text{virt}\to\text{real}} = \frac{\prod\limits_{(t,l)\in x\cup\{(\tilde{t},\tilde{l})\}} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)}{\prod\limits_{(t,l)\in x} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)} \times \frac{\prod\limits_{(t,l)\in x\cup\{(\tilde{t},\tilde{l})\}} \phi_{a_l'(t,l),l}(t-a_t'(t,l))}{\prod\limits_{(t,l)\in x} \phi_{a_l(t,l),l}(t-a_t(t,l))} \times \frac{\prod\limits_{(t,l)\in\tilde{x}\cup\tilde{c}'\setminus\{(\tilde{t},\tilde{l})\}} \kappa \cdot \phi_{a_l'(t,l),l}(t-a_t'(t,l))}{\prod\limits_{(t,l)\in\tilde{x}} \kappa \cdot \phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$= \frac{\exp\left(-(\kappa+1)\Phi_{\tilde{l},\star}(T-\tilde{t})\right) \phi_{a_l'(\tilde{t},\tilde{l}),\tilde{l}}(\tilde{t}-a_t'(\tilde{t},\tilde{l}))}{\kappa \cdot \phi_{a_l(\tilde{t},\tilde{l}),\tilde{l}}(\tilde{t}-a_t(\tilde{t},\tilde{l}))} \times \prod\limits_{(t,l)\in\tilde{c}'} \kappa \cdot \phi_{a_l'(t,l),l}(t-a_t'(t,l))$$

$$= \frac{\exp\left(-(\kappa+1)\Phi_{\tilde{l},\star}(T-\tilde{t})\right) \prod\limits_{(t,l)\in\tilde{c}'} \kappa \cdot \phi_{\tilde{l},l}(t-\tilde{t})}{\kappa}$$

Again, a number of terms cancel between $\mathcal{Q}_2^{\text{virt}\to\text{real}}$ and $\mathcal{L}_2^{\text{virt}\to\text{real}}$ to produce

$$\mathcal{A}_2^{\text{virt}\to\text{real}} = \frac{\exp(-\Phi_{\tilde{l},\star}(T-\tilde{t}))}{\kappa} \cdot \frac{n+\tilde{n}}{n+\tilde{n}+|\tilde{c}'|}$$

Changing a real event at time t with label l into a virtual event is entirely symmetric and produces the reciprocal, except that the total number of events with the "to be removed" children is $n+\tilde{n}$, and the resulting total number of events (after the removal) is $n+\tilde{n}-|\tilde{c}_i|$:

$$\mathcal{A}_2^{\text{real}\to\text{virt}} = \frac{\kappa}{\exp(-\Phi_{l,\star}(T-t))} \cdot \frac{n+\tilde{n}}{n+\tilde{n}-|\tilde{c}_i|} \;.$$

Move 3: Parent Sampling a new parent is the most difficult, as we may change whether the old, new, or both parents are virtual. Doing so will insert or remove virtual children (of the old or new real events). Let $A \in \{1+, 0, \text{v}\}$ be the status of the old parent after the proposed move: has children, has no children but is real, is virtual (respectively). Let $A' \in \{1+, 0, \text{v}\}$ be the status of the new parent before the proposed move.

Let (t, l) be the point to be changed. Let (t_p, l_p) be the old parent and (t_p', l_p') be the new parent. Let $\tilde{c}_p$ be the virtual children of the old parent. Let $\tilde{c}'$ be the new virtual children of the new parent, if the new parent was previously virtual. Finally, let δ^+ be the set of events moved from virtual to real, δ^- be the set of events moved from real to virtual events, $\tilde{\delta}^+$ be the set of newly added virtual events, and $\tilde{\delta}^-$ be the set of removed virtual events. If we let $\mathbf{1}$ be the indicator function, these sets can be written as

$$\delta^+ = \begin{cases} \{(t'_p, l'_p)\} & \mathbf{1}_{A'=v} = 1 \\ \{\} & \mathbf{1}_{A'=v} = 0 \end{cases} \quad \delta^- = \begin{cases} \{(t_p, l_p)\} & \mathbf{1}_{A=v} = 1 \\ \{\} & \mathbf{1}_{A=v} = 0 \end{cases}$$

$$\tilde{\delta}^+ = \begin{cases} \tilde{c}' & \mathbf{1}_{A'=v} = 1 \\ \{\} & \mathbf{1}_{A'=v} = 0 \end{cases} \quad \tilde{\delta}^- = \begin{cases} \tilde{c}_p & \mathbf{1}_{A=v} = 1 \\ \{\} & \mathbf{1}_{A=v} = 0 \end{cases} .$$

Using the definitions

$$u = \exp\left(-\Phi_{l_p,\star}(T - t_p)\right) \quad u' = \exp\left(-\Phi_{l'_p,\star}(T - t'_p)\right)$$

$$W = \sum_{(t'',l'')\in H_t} w(t'', l'')$$

$$\Delta W = \mathbf{1}_{A'=v} \sum_{(t'',l'')\in H_t(\tilde{c}')} w(t'', l'') - \mathbf{1}_{A=v} \sum_{(t'',l'')\in H_t(\tilde{c}_p)} w(t'', l'')$$

$$r = \frac{W}{W+\Delta W} \cdot \frac{n+\tilde{n}}{n+\tilde{n}+\mathbf{1}_{A'=v}|\tilde{c}'|-\mathbf{1}_{A=v}|\tilde{c}_p|}$$

the proposal ratio is

$$\mathcal{Q}_3 = \frac{\frac{1}{n+\tilde{n}+|\tilde{\delta}^+|-|\tilde{\delta}^-|} \frac{\phi_{l_p,l}(t-t_p)}{\sum\limits_{(t',l')\in H_t \cup H_t(\tilde{\delta}^+)\setminus H_t(\tilde{\delta}^-)} \phi_{l',l}(t-t')} \left(\frac{\kappa}{\kappa+1}\right)^{\mathbf{1}_{A'=v}} \left(\frac{1}{\kappa+1}\right)^{\mathbf{1}_{A'=0}}}{\frac{1}{n+\tilde{n}} \frac{\phi_{l'_p,l}(t-t'_p)}{\sum\limits_{(t',l')\in H_t} \phi_{l',l}(t-t')} \left(\frac{\kappa}{\kappa+1}\right)^{\mathbf{1}_{A=v}} \left(\frac{1}{\kappa+1}\right)^{\mathbf{1}_{A=0}}}$$

$$\times \frac{\left(\exp\left(-\kappa\cdot\Phi_{l_p,\star}(T-t_p)\right) \prod\limits_{(t',l')\in\tilde{c}_p} \kappa\cdot\phi_{l_p,l'}(t'-t_p)\right)^{\mathbf{1}_{A=v}}}{\left(\exp\left(-\kappa\cdot\Phi_{l'_p,\star}(T-t'_p)\right) \prod\limits_{(t',l')\in\tilde{c}'} \kappa\cdot\phi_{l'_p,l'}(t'-t'_p)\right)^{\mathbf{1}_{A'=v}}}$$

$$= \frac{(n+\tilde{n}) \frac{\phi_{l_p,l}(t-t_p)}{W+\Delta W} \left(\frac{\kappa+1}{\kappa}\right)^{\mathbf{1}_{A=v}-\mathbf{1}_{A'=v}}}{\left(n+\tilde{n}+|\tilde{\delta}^+|-|\tilde{\delta}^-|\right) \frac{\phi_{l'_p,l}(t-t'_p)}{W} (\kappa+1)^{\mathbf{1}_{A'=0}-\mathbf{1}_{A=0}}}$$

$$\times \frac{\left(\exp\left(-\kappa\cdot\Phi_{l_p,\star}(T-t_p)\right) \prod\limits_{(t',l')\in\tilde{c}_p} \kappa\cdot\phi_{l_p,l'}(t'-t_p)\right)^{\mathbf{1}_{A=v}}}{\left(\exp\left(-\kappa\cdot\Phi_{l'_p,\star}(T-t'_p)\right) \prod\limits_{(t',l')\in\tilde{c}'} \kappa\cdot\phi_{l'_p,l'}(t'-t'_p)\right)^{\mathbf{1}_{A'=v}}}$$

$$= r\,\frac{\phi_{l_p,l}(t-t_p)\left(\frac{\kappa+1}{\kappa}\right)^{\mathbf{1}_{A=v}-\mathbf{1}_{A'=v}}}{\phi_{l'_p,l}(t-t'_p)\,(\kappa+1)^{\mathbf{1}_{A'=0}-\mathbf{1}_{A=0}}}$$

$$\times \frac{\left(\exp\left(-\kappa\cdot\Phi_{l_p,\star}(T-t_p)\right) \prod\limits_{(t',l')\in\tilde{c}_p} \kappa\cdot\phi_{l_p,l'}(t'-t_p)\right)^{\mathbf{1}_{A=v}}}{\left(\exp\left(-\kappa\cdot\Phi_{l'_p,\star}(T-t'_p)\right) \prod\limits_{(t',l')\in\tilde{c}'} \kappa\cdot\phi_{l'_p,l'}(t'-t'_p)\right)^{\mathbf{1}_{A'=v}}}$$

The likelihood ratio is

$$\mathcal{L}_3 = \frac{\frac{\phi_{l'_p,l}(t-t'_p)}{\phi_{l_p,l}(t-t_p)} \prod\limits_{(t,l)\in x\cup\delta^+\setminus\delta^-} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)\phi_{a'_l(t,l),l}(t-a'_t(t,l))}{\prod\limits_{(t,l)\in x} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)\phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$\times \frac{\prod\limits_{(t,l)\in\tilde{x}\cup\tilde{\delta}^+\cup\delta^-\setminus\tilde{\delta}^-\setminus\delta^+} \kappa\cdot\phi_{a'_l(t,l),l}(t-a'_t(t,l))}{\prod\limits_{(t,l)\in\tilde{x}} \kappa\cdot\phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$= \frac{\prod\limits_{(t,l)\in\delta^+} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)\phi_{a'_l(t,l),l}(t-a'_t(t,l))}{\prod\limits_{(t,l)\in\delta^-} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)\phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$\times \frac{\phi_{l'_p,l}(t-t'_p) \prod\limits_{(t,l)\in\tilde{\delta}^+\cup\delta^-} \kappa\cdot\phi_{a'_l(t,l),l}(t-a'_t(t,l))}{\phi_{l_p,l}(t-t_p) \prod\limits_{(t,l)\in\tilde{\delta}^-\cup\delta^+} \kappa\cdot\phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$= \frac{\phi_{l'_p,l}(t-t'_p) \prod\limits_{(t,l)\in\delta^+} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)}{\phi_{l_p,l}(t-t_p) \prod\limits_{(t,l)\in\delta^-} \exp\left(-(\kappa+1)\Phi_{l,\star}(T-t)\right)}$$

$$\times \frac{\frac{1}{\kappa} \prod\limits_{(t,l)\in\tilde{\delta}^+} \kappa\cdot\phi_{a'_l(t,l),l}(t-a'_t(t,l))}{\frac{1}{\kappa} \prod\limits_{(t,l)\in\tilde{\delta}^-} \kappa\cdot\phi_{a_l(t,l),l}(t-a_t(t,l))}$$

$$= \frac{\phi_{l'_p,l}(t-t'_p)\left(\exp\left(-(\kappa+1)\Phi_{l'_p,\star}(T-t'_p)\right)\frac{1}{\kappa}\right)^{\mathbf{1}_{A'=v}}}{\phi_{l_p,l}(t-t_p)\left(\exp\left(-(\kappa+1)\Phi_{l_p,\star}(T-t_p)\right)\frac{1}{\kappa}\right)^{\mathbf{1}_{A=v}}}$$

$$\times \frac{\left(\prod\limits_{(t,l)\in\tilde{c}'} \kappa\cdot\phi_{l'_p,l}(t-t'_p)\right)^{\mathbf{1}_{A'=v}}}{\left(\prod\limits_{(t,l)\in\tilde{c}_p} \kappa\cdot\phi_{l_p,l}(t-t_p)\right)^{\mathbf{1}_{A=v}}} .$$

Many terms from $\mathcal{Q}_3$ and $\mathcal{L}_3$ cancel. Using u' and u from above, we get

$$\mathcal{A}_3 = r\,\frac{\left(\exp\left(-\Phi_{l'_p,\star}(T-t'_p)\right)\frac{1}{\kappa}\right)^{\mathbf{1}_{A'=v}} (\kappa+1)^{\mathbf{1}_{A=v}-\mathbf{1}_{A'=v}}}{\left(\exp\left(-\Phi_{l_p,\star}(T-t_p)\right)\frac{1}{\kappa}\right)^{\mathbf{1}_{A=v}} (\kappa+1)^{\mathbf{1}_{A'=0}-\mathbf{1}_{A=0}}}$$

$$= r\,\frac{(u')^{\mathbf{1}_{A'=v}} (\kappa+1)^{\mathbf{1}_{A=v}+\mathbf{1}_{A=0}}}{(u)^{\mathbf{1}_{A=v}} (\kappa+1)^{\mathbf{1}_{A'=0}+\mathbf{1}_{A'=v}}}$$

which can be rewritten in tabular form, if we note that $r = 1$ unless $A' = v$ or $A = v$:

A\A’	1+	0	v
1+	1	$\frac{1}{\kappa+1}$	$\frac{ru'}{\kappa+1}$
0	$\frac{\kappa+1}{1}$	1	$\frac{ru'}{1}$
v	$\frac{r(\kappa+1)}{u}$	$\frac{r}{u}$	$\frac{ru'}{u}$

Algorithm 3 MCMC Algorithm, initialization and helper function

```
Initialize sampler by setting all evidence events to be children of the root event and sampling virtual children for them.
function MCMCInit($z$,$T$,$\kappa$)
  $x \leftarrow \{(0,0,())\} \cup z^{(o)}$
  for all $(t,l,a) \in x$
    $\tilde{x} \leftarrow \tilde{x} \cup$ SampleChildren($l_i$,$t_i$,$T$,$z^{(o)}$,$\kappa$)
  return $(x,\tilde{x})$

Sample children for event ($t_i$,$l_i$) for all labels until time $T$, inserting parent information
function SampleChildren($l_i$,$t_i$,$T$,$z^{(o)}$,$\kappa$)
  $c \leftarrow$ SampleWithEvid($l_i$,$t_i$,$T$,$z^{(o)}$,$\kappa$) , $r \leftarrow \{\}$
  for $(t,l) \in c : r \leftarrow r \cup (t,l,(t_i,l_i))$
  return $r$
```

The pseudo-code for the MCMC sampler is listed in Algorithms 3 and 4. For the purposes of this pseudo-code, we augment each element of x and $\tilde{x}$ with an extra value: the parent, which we denote by its (t,l) pair. In actual code, references would be used instead. We initialize the sampler with just the root event and the events from the evidence and set all events to have the root event as their parents. This assumes that all labels have a non-zero background rate. If not, a more complex initialization would be necessary. In our implementation, a few indexes make things faster. Additionally, when resampling the parent for event of label l', we exclude all events for labels l for which $M_{l,l'} = 0$. For sparse systems, this saves much time. Similar optimizations for sparse systems are applied where possible in our implementation of both methods.

Synthetic Results

For each kernel, we tested the algorithms on two sizes of problems, each with an “easy” and “hard” version.

Small Problem In our small problems, the labels form a chain. For the simple version there are three labels, and M has only two non-zero entries: $M_{1,2} = M_{2,3} = 1$. So, each label affects the next one in the chain but there are no self-excitations. The base rates are 10^{-2} for label 1 and 10^{-6} for labels 2 and 3. $T = 3.0$. Labels 1 and 2 are unobserved the entire time. Label 3 is observed only on $[2.0, 3.0)$. The only observed event is $(t = 2.5, l = 3)$. We query the (expected) number of events of label 1. For the exponential kernel, the true value is ≈ 1.027. For the power-law kernel, the true value is ≈ 1.026.

The hard version consists of 5 labels. Each label affects the next two in the chain (but not itself): $M_{l,l+1} = M_{l,l+2} = 1$.

Algorithm 4 MCMC Algorithm

```
Sampler step, given previous state ($x$, $\tilde{x}$), evidence ($z$), ending time $T$, and $\kappa$
function MCMCStep($x$,$\tilde{x}$,$z$,$T$,$\kappa$)
  $(t,l,(t_p,l_p)) \leftarrow$ UniformSamp($x \cup \tilde{x}$)
  $N \leftarrow |x| + |\tilde{x}|$                    ▷ total number of events
  $\tilde{c} \leftarrow \{(t',l',a) \in \tilde{x} \mid a = (t,l)\}$     ▷ all virtual children
  switch UniformSamp({move-1, move-2, move-3})
    case move-1
      $\tilde{c}' \leftarrow$ SampleChildren($t$,$l$,$T$,$z^{(o)}$,$\kappa$)
      if Rand(0,1) $< N/(N + |\tilde{c}'| - |\tilde{c}|)$
        $\tilde{x} \leftarrow \tilde{x} \cup \tilde{c}' \setminus \tilde{c}$
    case move-2
      if $(t,l,(t_p,l_p)) \in x \wedge (t,l) \notin z^{(x)} \wedge l \neq 0 \wedge |c| = 0$
        $r \leftarrow (\kappa/\exp(-\Phi_{l,\star}(T-t)))(N/(N-|\tilde{c}|))$
        if Rand(0,1) $< r$
          $x \leftarrow x \setminus (t,l,(t_p,l_p))$
          $\tilde{x} \leftarrow \tilde{x} \cup (t,l,(t_p,l_p)) \setminus c$
      else if $(t,l,(t_p,l_p)) \in \tilde{x}$
        $\tilde{c}' \leftarrow$ SampleChildren($t$,$l$,$T$,$z^{(o)}$,$\kappa$)
        $r \leftarrow (\exp(-\Phi_{l,\star}(T-t))/\kappa)(N/(N+|\tilde{c'}|))$
        if Rand(0,1) $< r$
          $\tilde{x} \leftarrow \tilde{x} \cup \tilde{c}' \setminus (t,l,(t_p,l_p))$
          $x \leftarrow x \cup (t,l,(t_p,l_p))$
    case move-3
      if $(t,l,(t_p,l_p)) \notin \tilde{x}$
        $w \leftarrow \{\}$
        for all $(t',l',a) \in (x \cup \tilde{x}) \mid t' < t$
          $w \leftarrow w \cup (\exp(-\Phi_{l',\star}(T-t')) : t',l',a)$
        $(w' : t'_p, l'_p, a) \leftarrow$ WtSetSample($w$)
        if $(t'_p, l'_p, a) \in \tilde{x}$
          $\tilde{c}'_p \leftarrow$ SampleChildren($t'_p$,$l'_p$,$T$,$z^{(o)}$,$\kappa$)
        $v \leftarrow$ false
        if $(t_p,l_p)$ can be made virtual after change
          $v \leftarrow$ Rand(0,1) $< \kappa/(\kappa+1)$
        if Rand(0,1) < ratio from table
          $x \leftarrow x \cup (t,l,(t'_p,l'_p)) \setminus (t,l,(t_p,l_p))$
          if $(t'_p, l'_p, a) \in \tilde{x}$
            $\tilde{x} \leftarrow \tilde{x} \cup \tilde{c}'_p \setminus (t'_p, l'_p, a)$
            $x \leftarrow x \cup (t'_p, l'_p, a)$
          if $v$
            $\tilde{c}_p \leftarrow \{(t',l',a) \in \tilde{x} \mid a = (t_p,l_p)\}$
            $(t_{pp},l_{pp}) \leftarrow (t'',l'') \mid (t_p,l_p,(t'',l'')) \in x$
            $x \leftarrow x \setminus (t_p,l_p,(t_{pp},l_{pp}))$
            $\tilde{x} \leftarrow \tilde{x} \cup (t_p,l_p,(t_{pp},l_{pp})) \setminus \tilde{c}_p$
  return $(x,\tilde{x})$
```

The base rates are 10^{-3} for label 1 and 10^{-6} for the others. We sampled from the model (with different samples for each kernel, but fixing a single sample for all experiments with a given kernel) with $T = 1000$. The observations were of only labels 3 and 5 (for all time). We queried the number of events for label 4. The true values are ≈ 44.88 for the exponential kernel and ≈ 12.90 for the power-law kernel.

Large Problem In this model, the connections between labels are generated by a randomly sampled connected undirected graph with degrees sampled from a power-law distribution with $\alpha = 2.5$ (Viger and Latapy, 2016). This simulates common social network graph structures, a common application of Hawkes processes. We use a single sampled 100-node graph for the easy problem and 500-node graph for the hard problem. Each node in the graph is associated with a label. If there is not a direct edge between nodes i and j, $M_{i,j} = 0$. If there is an edge, $M_{i,j} = 1/(8d_j)$ where d_j is the degree of node j (thus the graph has asymmetric weights). $M_{i,i} = 1/2$. The base rates are 0.1 for all labels. Separately for each kernel, a single full event sequence is sampled. The labels are sorted according to degree and alternating labels are completely unobserved (other labels are complete observed). We query the total number of unobserved events.

For the easy version, $T = 10$. The true values are ≈ 91.44 for the exponential kernel and ≈ 79.11 for the power-law kernel. For the hard version, $T = 100$. The true values are ≈ 4739.6 for the exponential kernel and ≈ 4473.0 for the power-law kernel.

MCEM

Assuming complete data (or a set of samples generated from the posterior distribution) and assuming the auxiliary information of which event "caused" which event (a), we can maximize the likelihood in almost closed form. We derive the method for a single sample below (for ease of notation). To extend to multiple samples, sum the relevant statistics over all samples. Note: a_i is the index of the parent of the ith event and l_i is the label of the ith event. For events "caused" by the background rate, we say that the label of their parent event is 0 (the special label of the root event). ν is the strength of the L_1 regularizer (0 if no regularization). Finally, we assume the kernel takes the form $\phi_{l',t}(l) = M_{l,l'}\phi(t)$ and therefore denote the integral of the kernel as $\Phi_{l',t}(l) = M_{l,l'}\Phi(t)$.

$$\begin{aligned}
L = &-\sum_i \left(\sum_l M_{l_i,l}\right)\Phi(T-t_i) - \sum_l T\mu_l \\
&+ \sum_{i|l_{a_i}\neq 0}\left(\log M_{l_{a_i},l_i} + \log\phi(t_i - t_{a_i})\right) \\
&+ \sum_{i|l_{a_i}=0}\log\mu_{l_i} - \nu\sum_l\sum_{l'} M_{l,l'}
\end{aligned}$$

The maximizing value for μ_l is immediate:

$$\begin{aligned}
\frac{\partial L}{\partial \mu_l} &= 0 \\
T - \frac{\sum_{i|l_{a_i}=0,l_i=l} 1}{\mu_l} &= 0 \\
\mu_l &= \frac{N_{0,l}}{T}
\end{aligned}$$

where $N_{0,l}$ is the number of events of label l generated from the root event.

The maximizing value for $M_{j,k}$ can be derived, assuming the base kernel parameters are fixed:

$$\begin{aligned}
&\frac{\partial L}{\partial M_{j,k}} = 0 \\
&\quad 0 = -\sum_{i|l_i=j}\Phi(T-t_i) + \frac{\sum_{i|l_i=k,l_{a_i}=j} 1}{M_{j,k}} - \nu \\
&M_{j,k} = \frac{N_{j,k}}{\nu + \sum_{i|l_i=j}\Phi(T-t_i)}
\end{aligned}$$

where $N_{j,k}$ is the number of events of label k generated from events of label j.

Substituting the maximizing value of M into L (and ignoring the terms that do not depend on the kernel), our goal is to select the kernel parameter(s) that maximize(s) the expression

$$\begin{aligned}
L = &-\sum_i \left(\sum_l \frac{N_{l_i,l}}{\nu + \sum_{j|l_j=l_i}\Phi(T-t_j)}\right)\Phi(T-t_i) \\
&- \sum_{i|l_{a_i}\neq 0}\log\left(\nu + \sum_{j|l_j=l_{a_i}}\Phi(T-t_j)\right) \\
&+ \sum_{i|l_{a_i}\neq 0}\log\phi(t_i - t_{a_i}) \\
&- \nu\sum_l\sum_{l'}\frac{N_{l,l'}}{\nu + \sum_{i|l_i=l}\Phi(T-t_i)}\ .
\end{aligned}$$

If we let $S_l = \sum_{i|l_i=l}\Phi(T-t_i)$ and $R = \sum_{i|l_{a_i}\neq 0}\log\phi(t_i - t_{a_i})$, we can simplify the expression to

$$\begin{aligned}
L = &-\sum_l\sum_{i|l_i=l}\sum_{l'}\frac{N_{l,l'}}{\nu + S_l}\Phi(T-t_i) \\
&- \sum_{i|l_{a_i}\neq 0}\log\left(\nu + S_{l_{a_i}}\right) + R - \nu\sum_l\sum_{l'}\frac{N_{l,l'}}{\nu + S_l} \\
= &\sum_l\sum_{l'}\frac{N_{l,l'}}{\nu + S_l}\left(\nu + \sum_{i|l_i=l}\Phi(T-t_i)\right) \\
&- \sum_{i|l_{a_i}\neq 0}\log\left(\nu + S_{l_{a_i}}\right) + R \\
= &\, N - \sum_{i|l_{a_i}\neq 0}\log\left(\nu + S_{l_{a_i}}\right) + R
\end{aligned}$$

where N is a constant (the number of events not generated from the root event). S_l and R are functions of the base kernel (and therefore its parameters). This expression can therefore be optimized by a low-dimensional search on the base kernel parameters. Once the base kernel parameters are fixed, M can be estimated as above.

Additional Network Analysis

The networks shown in Figure 4, left (main paper) appear very messy. We tried a number of procedures to tease out clusters or meaningful graphs. The best procedure we found was the following.

1. We filtered out edges with weight less than 0.002 (threshold found by manual search).
2. We removed nodes with small degree (less than 5).
3. We used a community detection algorithm based on modularity (Blondel et al., 2008) to get four or five clusters.
4. We then used a graph layout algorithm to position the nodes based on both links and the detected clusters.
5. We plotted the graph with labels indicating the Chicago community numbers (numbers greater than 77 indicate hidden labels) with node colors corresponding to the clusters and sizes corresponding to the degrees.

The results (along with geographic maps of Chicago showing the clusterings) are shown in Figure 4, middle & right (main paper). The increased sparsity when using hidden labels is apparent. While the clusters detected are not unreasonable geographically, they do not reflect the major connectivity of the model. Note that the strongest edges (thickest lines) exist *between* clusters, *not within* clusters, particularly in the model without hidden labels. The hidden labels (77 and 80) can be seen as major nodes in the right-most graph.

References

Bacry, E.; Mastromatteo, I.; and Muzy, J.-F. 2015. Hawkes processes in finance. Market Microstructure and Liquidity 1(1).

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment P10008.

Viger, F., and Latapy, M. 2016. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. Journal of Complex Networks 4(1):15–37.