Point-process models of social network interactions: parameter estimation and missing data recovery

JOSEPH R. ZIPKIN[1], FREDERIC P. SCHOENBERG[2], KATHRYN CORONGES[3], and ANDREA L. BERTOZZI[4]

[1] *Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095, USA*
email: jzipkin@math.ucla.edu
[2] *Department of Statistics, University of California, Los Angeles, Los Angeles, CA 90095, USA*
[3] *Department of Behavioral Sciences and Leadership, United States Military Academy, West Point, NY 10996, USA*
[4] *Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095, USA*

(*Received August 10, 2014*)

Electronic communications and other categories of interactions within social networks exhibit bursts of activity localized in time. We adopt a self-exciting point process model for such activities. We discuss parameter estimation for such point processes. We then present a method for filling in missing data in records of electronic communications and demonstrate the method using a data set composed of email records. The ability to fill in large blocks of missing social network data has implications for security, surveillance, and privacy.

1 Introduction

1.1 Burstiness and Hawkes processes

The ways humans interact has long been a subject of interest. The rise of electronic communication, and particularly social media, has made large data sets of human interactions available. Growing interest in privacy and cybercommunications has led to questions about what can be learned from this data and how it is used.

A natural first question is how to model patterns of social interactions. A point process seems a natural choice, but the simplest point process, the Poisson process, is ill suited to modeling several classes of human activity, including communication. The problem, broadly speaking, is that human activity patterns tend to be "bursty", that is, well clustered in time relative to a Poisson process. See, for example, Figure 1. Two time series are plotted. Figure 1(a) is taken from the IkeNet data set, of which more later. It shows the times that two particular users sent each other emails. Figure 1(b) is a realization of a Poisson process. The two time series have the same number of events, but the IkeNet time series clearly better clustered. This suggests a Poisson process is a suboptimal choice for modeling human interactions. Bursty dynamics have been observed in Web browsing [28], emails [1], communications within electronic social networking systems [26], mobile phone calls [18], FTP requests [24], and even face-to-face interactions [12].

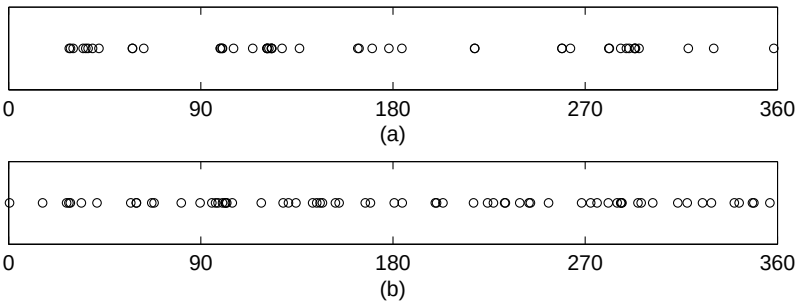

FIGURE 1. Two time series. The axis is time, and circles indicate events. Each time series has 68 events. (a) Timestamps of emails sent between IkeNet user 6 and IkeNet user 15. (b) A simulated Poisson process.

In 1971 Hawkes [9, 10] introduced a class of *self-exciting point processes* that have come to bear his name. A *Hawkes process* is a nonhomogeneous Poisson process $n(t)$ whose intensity is governed by

$$\lambda(t) = \mu + \sum_{t_i<t} g(t - t_i; \theta). \tag{1.1}$$

Each t_i is an event time, μ is a deterministic *background intensity*, and g is a *triggering function* specifying how much a recent event increases the intensity, hence the notion of the Hawkes process as self-exciting. Here we note explicitly the dependence of g on a vector θ of parameters because we will estimate these parameters statistically, but we may omit it later for notational convenience. (Nonparametric approaches to estimating g have also been developed [13, 16].) Likewise we may write $\lambda(t|\{t_i\}_{i=1}^{n(t)})$ when we want to emphasize the dependence of λ on the history. The background intensity μ can be time-dependent, but we take it as a constant for simplicity.

Figure 2 shows Hawkes process realizations with $\mu = 0.15$ and $g(t) = 0.5e^{-0.6t}$. The intensity and event times are plotted against time. The Hawkes process events are more tightly clustered in time than the Poisson process of Figure 1(b), perhaps more closely resembling Figure 1(a).

The Hawkes process appears in the seismology literature as a model for the timing of earthquakes and their aftershocks [20]. As interest in and availability of large data sets of human activities have grown, Hawkes processes have been used to model electronic communications [4], gang crimes [7, 11, 27], and even terrorist and insurgent activity [14, 19].

The constraints on μ and g are modest. First, we assume that $\mu > 0$. Second, so that the process is self-exciting rather than self-dampening, we assume g is non-negative. Finally, we assume that $\int_0^\infty g(t;\theta)dt < 1$ to ensure that the process is stationary. The importance of this assumption becomes clear when we recognize that $\int_0^\infty g(t;\theta)dt$ is the expected number of immediate descendants of each event. Were it greater than 1, then each event could be expected to give rise to infinitely many others. This would make the process explosive and impossible to simulate repeatedly. It also runs agains intuition for our application to emails within a social network (all email threads end eventually) or indeed any of the other applications mentioned above.

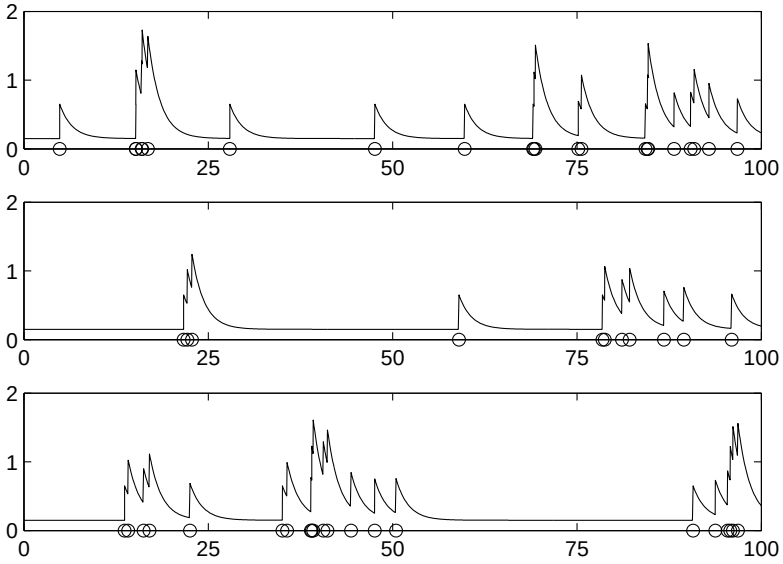

FIGURE 2. Three realizations of a Hawkes process with $\mu = 0.15$ and $g(t) = 0.5e^{-0.6t}$. The horizontal axis is time. Circles indicate events, and the solid curve is the intensity.

Our approach recalls that of Stomakhin, Short & Bertozzi's work on networks of criminal gang rivalries [27]. A gang that has been victimized by a rival will often retaliate, setting off a burst of tit-for-tat crimes. Stomakhin, Short & Bertozzi associate to each pair of rival gangs an independent Hawkes process whose events represent crimes committed by one gang against the other. Then, noting that law enforcement often knows which gang was victimized but not which gang was the perpetrator, they cast the task of solving the crime as a missing data problem, in which a history of gang crimes is known but some of the identities of the gangs involved in particular incidents are hidden. Like Stomakhin, Short & Bertozzi, we will assign independent Hawkes processes to the connections within a social network and solve a missing data problem. However, our variational approach will be different.

1.2 The IkeNet data set

Between 2010 and 2011, as part of the United States Military Academy's IkeNet research program on social networks, researchers collected email data from 22 volunteers among officers and cadets affiliated with the Academy. The data set consists of time stamps and anonymized sender and receiver codes from 8,896 emails sent among these volunteers over a 361-day period. This is a social network with 253 connections. (We include self-connections because the volunteers emailed themselves.) Emails were sent along 250 of these connections.

The emails are by no means distributed evenly among these 250 connections. Table 1 lists the 12 pairs of cadets who exchanged more than 100 emails. The top pair (9,18)

Table 1. *Pairs of cadets who exchanged* > 100 *emails*

Pair	Number of emails	Pair	Number of emails
(9,18)	1,042	(18,22)	222
(11,22)	511	(4,13)	134
(13,17)	302	(9,13)	131
(11,13)	293	(13,18)	130
(8,18)	281	(13,22)	120
(13,15)	223	(3,17)	116

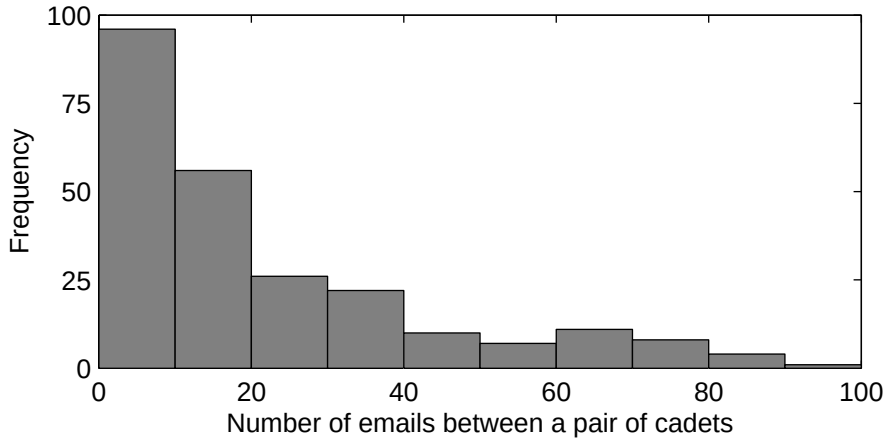

FIGURE 3. Histogram of the number of emails sent between each pair of cadets. Only pairs who exchanged fewer than 100 emails are shown; see Table 1 for the others.

exchanged 1,042 emails, or 11.7% of all the emails in the corpus. Together these top 12 exchanged 3,505 emails, or 39.4% of the corpus. Figure 3 is a histogram of the number of emails exchanged among the remaining pairs, all of them less than 100. Many of the pairs of cadets exchanged only a few emails, while a few pairs exchanged a substantial proportion of all emails in the corpus, and a few users (13, 18, 22) appear three times or more in this list of highly active pairs. These observations are consistent with a core–periphery structure, which is a characteristic of many social networks [6].

Fox *et al.* [8] perform several statistical studies of this data set, including fitting Hawkes processes to the email patterns via maximum likelihood estimation. They find that a Hawkes process model fits the IkeNet data better than a homogeneous Poisson model, as measured by the Akaike information criterion (AIC). They also incorporate the results of a leadership survey administered to the volunteers, revealing more details of the social network.

Our approach differs from Fox *et al.*'s in two basic ways. First, while they assign an independent Hawkes process to each cadet (i.e., each node in the network), we assign one to each relationship between cadets (i.e., each edge in the network). This is appropriate to the missing data problem, in which differences in the cadets' relationships matter a great deal. Second, while Fox *et al.* allow the background rate μ to change periodically to

capture daily and weekly rhythms in email traffic, we take μ as a constant. We expect this simplification's impact to be modest, because Fox *et al.* found only a modest improvement in AIC by moving to a time-varying μ, and because we do not expect it to have much import for our missing data problem. Taking a constant μ has precedent in seismology [16].

2 EM estimation of Hawkes process parameters

First we must discuss fitting the parameters of a Hawkes process to data. We take a maximum-likelihood approach, using an expectation-maximization numerical method to combat the problem's ill conditioning [16, 29]. Finally, we give several examples for different choices of the triggering function g. It is most common in the literature to assume an exponential form for g [4, 8, 11, 17, 27], though other forms are also in use, including power law [5, 21] and the exponential multiplied by a polynomial [22]. Our comparison of exponential and power-law forms suggests that it does not matter which is used, validating the frequent use of the exponential form.

The general problem is, given an interval $[0, T]$ and a time series $\{t_i\}_{i=1}^{n(T)}$ falling in that interval, to produce statistical estimates $\hat{\mu}$ and $\hat{g}$ for the μ and g of the Hawkes process assumed to generate the data. Nonparametric methods of estimating g exist [13], but our approach will be to assume a form for g (in statistical parlance, to adopt a *model* for g) and instead estimate θ, the vector of parameters, together with μ using maximum likelihood, yielding parameter estimates $(\hat{\mu}, \hat{\theta})$.

The likelihood that a nonhomogeneous Poisson process generated a history $\{t_i\}_{i=1}^{n(T)}$ is

$$L = \exp\left(- \int_0^T \lambda(t|\{t_i\}_{i=1}^{n(T)})dt \right) \prod_{i=1}^{n(T)} \lambda(t_i|\{t_j\}_{j=1}^{i-1}). \tag{2.1}$$

See [25] for a detailed discussion. It is standard to instead maximize the log-likelihood, which for a Hawkes process as in (1.1) has the form

$$\log L(\mu, \theta) = \sum_{i=1}^{n(T)} \left(\log\left(\mu + \sum_{j=1}^{i-1} g(t_i - t_j; \theta) \right) - \int_0^{T-t_i} g(t; \theta)dt \right) - \mu T. \tag{2.2}$$

Ozaki [23] treats maximum likelihood estimation of the parameters when g is exponential. The likelihood function can be ill conditioned, so optimization techniques must be chosen with care.

2.1 Generating Hawkes process time series

Throughout this section, and again in section 4 when considering simulated networks, we use Lewis's thinning method [15, 20] to generate artificial Hawkes process time series. Briefly, given a history $\{t_i\}_{i=1}^{n}$ at time t, we simulate an independent exponential random variable s with rate parameter $\lambda(t|\{t_i\}_{i=1}^{n})$. Were this process homogeneous, we would take $t_{n+1} = t + s$, set $t = t + s$, and continue. However, because the intensity decays following an event, we only do this with probability $\lambda(t + s|\{t_i\}_{i=1}^{n})/\lambda(t|\{t_i\}_{i=1}^{n})$. If we do not, we set $t = t + s$ and generate a new s. The procedure continues until $t > T$.

2.2 The EM algorithm

To estimate the Hawkes process parameters we adapt the expectation-maximization (EM) algorithm of Veen & Schoenberg [29]. The algorithm maximizes the likelihood (2.1), but indirectly, so as to avoid the conditioning problems of maximizing (2.2) by standard iterative methods.

The algorithm relies on the Hawkes process's branching structure. The linearity of the intensity process (1.1) allows us to calculate the probability that a given event was triggered by any previous event; otherwise it is a *background* event. The probability that an event occurring at time t_i is a background event is $\mu/\lambda(t_i)$, and the probability that it was caused by an event that occurred at time $t_j < t_i$ is $g(t_i - t_j)/\lambda(t_i)$.

The EM algorithm alternates between an *expectation step* and a *maximization step*. At the k^{th} iteration we have an estimate $(\mu^{(k)}, \theta^{(k)})$ of the parameters. The expectation step of the $(k+1)^{\text{th}}$ iteration uses those parameters to calculate $p_{i,i}^{(k+1)}$ and $p_{i,j}^{(k+1)}$, respectively the probabilities that event i was a background event or was caused by event j:

$$p_{i,i}^{(k+1)} = \frac{\mu^{(k)}}{\mu^{(k)} + \sum_{j=1}^{i-1} g(t_i - t_j; \theta^{(k)})},$$

$$p_{i,j}^{(k+1)} = \frac{g(t_i - t_j; \theta^{(k)})}{\mu^{(k)} + \sum_{j=1}^{i-1} g(t_i - t_j; \theta^{(k)})}.$$

The maximization step maximizes the *complete data likelihood* of the branching structure. The likelihood of a given structure can be decomposed into independent pieces:

- The number of background events. This is a Poisson random variable (call it b) with expectation μT. Its likelihood is

$$L_1(\mu) = e^{-\mu T} \frac{(\mu T)^b}{b!}.$$

- The number of immediate descendants of each event, both background and triggered, given b. Let d_i be the number of descendants of event i. It is also Poisson, and its expectation is $\int_0^{T-t_i} g(t;\theta)dt$. Lewis & Mohler [13] found that approximating this by $G(\theta) = \int_0^{\infty} g(t;\theta)dt$ had only a modest impact on the reliability of results, so we adopt this approximation for simplicity. Because each d_i is independent of the others, their joint likelihood is

$$L_2(\theta) = \prod_{i=1}^{n} e^{-G(\theta)} \frac{G(\theta)^{d_i}}{d_i!}.$$

- The timing of the descendant events given b and all the d_i. Let $j(i)$ be the event of which i is the immediate descendant, with $j(i) = i$ if i is a background event. The likelihood of event i occurring at time t_i is $g(t_i - t_{j(i)};\theta)/G(\theta)$ (we again approximate a finite integral of g by $G(\theta)$), so the joint likelihood of all events' timing is

$$L_3(\theta) = \prod_{i:j(i)<i} \frac{g(t_i - t_{j(i)};\theta)}{G(\theta)}.$$

The background events are distributed uniformly in $[0, T]$, so their timing does not enter into the likelihood.

The likelihood of the overall branching structure is the product of $L_1(\theta)$, $L_2(\theta)$, and $L_3(\theta)$. The log-likelihood is

$$\ell_c(\mu,\theta) = -\mu T + b\log\mu + b\log T - \log(b!) + \sum_{i=1}^{n}(-G(\theta) + d_i \log G(\theta) - \log(d_i!)) + \sum_{i:j(i)<i} (\log g(t_i - t_{j(i)};\theta) - \log G(\theta)).$$

We are maximizing with respect to the parameters (μ,θ), so we disregard additive terms that are constants in them. Then we take the expectation with respect to the probabilities calculated in the expectation step:

$$E^{(k+1)}(\mu,\theta) = -\mu T + (\log\mu)\sum_{i=1}^{n} p_{i,i}^{(k+1)} - nG(\theta) + \sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k+1)} \log g(t_i - t_j;\theta).$$

It is this function that we maximize with respect to (μ,θ).

Regardless of the model for g, the maximizing value of μ is

$$\hat{\mu}^{(k+1)} = \frac{\sum_{i=1}^{n} p_{i,i}^{(k+1)}}{T}.$$

The maximizing θ satisfies

$$\nabla G(\hat{\theta}^{(k+1)}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k+1)} \frac{\nabla_\theta g(t_i - t_j;\hat{\theta}^{(k+1)})}{g(t_i - t_j;\hat{\theta}^{(k+1)})}. \tag{2.3}$$

Fortunately, for both the models we choose for g, (2.3) reduces to tractable algebraic expressions for each component of $\hat{\theta}^{(k+1)}$.

2.3 Example: exponential triggering

First, we choose $g(t;\alpha,\omega) = \alpha\omega e^{-\omega t}$. The L^1 condition on g is equivalent to $\omega > 0$ and $0 \le \alpha < 1$. The θ condition (2.3) reduces to

$$\hat{\alpha}^{(k+1)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k)}}{n}, \qquad \hat{\omega}^{(k+1)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k)}}{\sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k)}(t_i - t_j)}.$$

We generated 50,000 realizations of a Hawkes process with this triggering function, with $T = 361$, $\mu = 0.05$, $\alpha = 0.5$, and $\omega = 6$. (These values were chosen to correspond with the IkeNet data.) We then estimated the parameters using the EM algorithm. The results are presented in Table 2 and Figure 4(a). The estimates for the parameters are distributed about their ground-truth values, with a slight rightward skew for μ and more pronounced leftward and rightward skews for α and ω, respectively. Of the 50,000 estimates for ω, 504 or about 1% were greater than 18; these are omitted from the histogram.

2.4 Example: power-law triggering

Many human behavior patterns exhibit power-law scaling in inter-event times [1]. Therefore, we now choose $g(t;\alpha,q) = \alpha(q-1)(1+t)^{-q}$. This has the same number of parameters

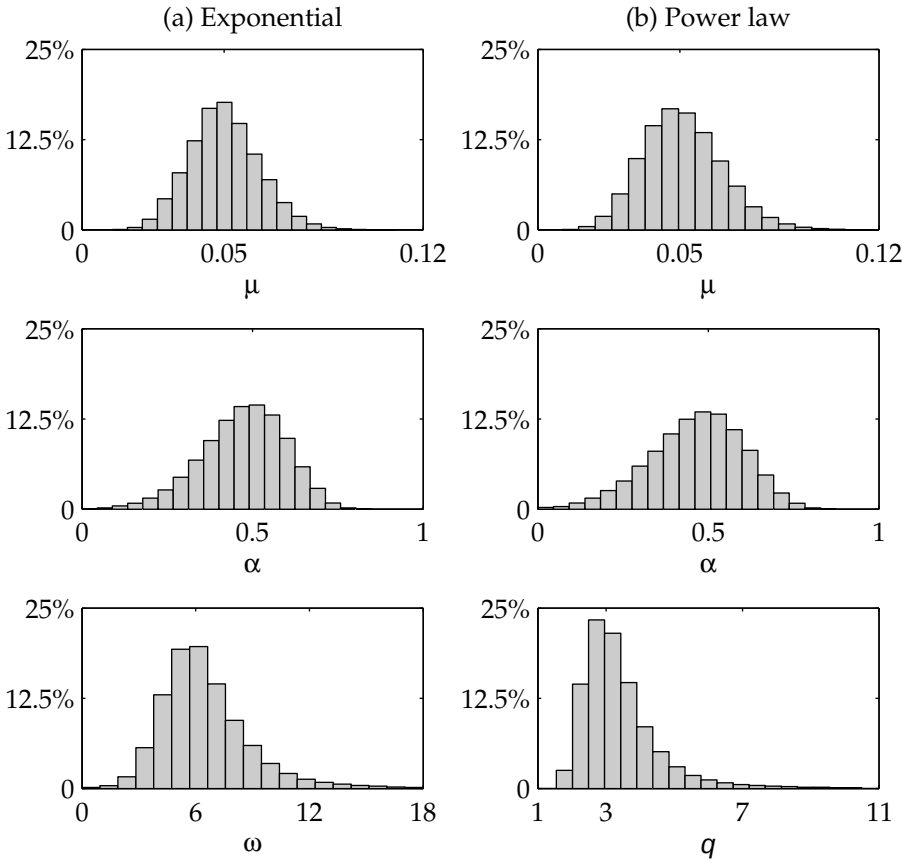

FIGURE 4. Histograms showing the results of EM estimation of model parameters for (a) exponential and (b) power law triggering functions. For each model 50,000 time series were generated. About 1% of the results for ω and q are omitted because they are outliers that exceed the right limit of the graph.

Table 2. *EM estimation results*

Model	Parameter	Ground truth	Mean
Exponential	μ	0.05	0.05002
	α	0.5	0.4733
	ω	6	6.753
Power law	μ	0.05	0.05095
	α	0.5	0.4641
	q	3	3.590

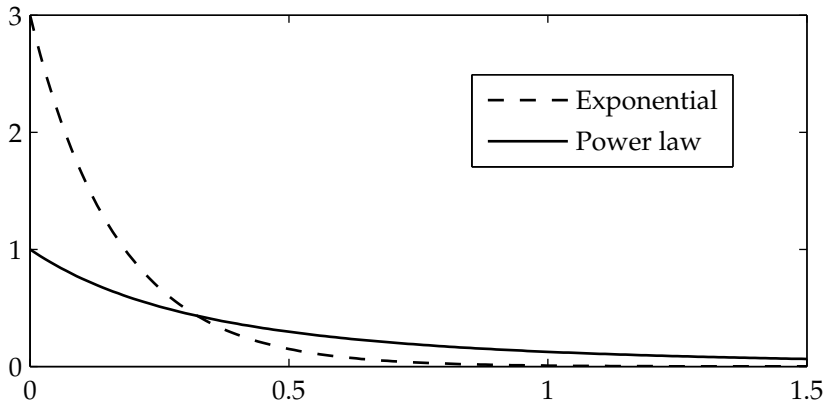

FIGURE 5. Triggering functions. Exponential: $g(t) = 3e^{-6t}$. Power law: $g(t) = (1+t)^{-3}$.

as the previous section's exponential model. The L^1 condition on g is equivalent to $q > 1$ and $0 \leq \alpha < 1$. The θ condition (2.3) reduces to

$$\hat{\alpha}^{(k+1)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k)}}{n}, \qquad \hat{q}^{(k+1)} = 1 + \frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k)}}{\sum_{i=1}^{n}\sum_{j=1}^{i-1} p_{i,j}^{(k)} \log(1 + t_i - t_j)}.$$

Again, we generated 50,000 realizations with T, μ, and α as above, and $q = 3$. The results are presented in Table 2 and Figure 4(b). As with the exponential triggering function, estimates for μ and α are overall close to their ground truths with, respectively, a slight rightward skew and a more pronounced leftward skew,. The estimates of q clearly peak around 3 but skew rightward. Of the 50,000 estimates for q, 446 or about 0.9% were greater than 11; these are omitted from the histogram.

2.5 Comparison of exponential and power-law

In practice we may not know the best form of the triggering function to use when modeling a point process. Nonparametric methods are one solution [13]; however, these can be cumbersome, and without enough data they invite overfitting. Instead we ask whether time series generated by the two triggering functions discussed in sections 2.3 and 2.4 can be told apart. The triggering functions are plotted together in Figure 5. They have the same integral, but the power-law triggering function has a longer tail. One might reasonably expect these two triggering functions to produce different behaviors.

Most of the time we consider the likelihood only in the context of maximizing it with respect to the parameters or the model, given a history. But the likelihood has comparative value, as well. Comparing the likelihoods of models or sets of parameters to the maximum likelihood value reveals how much likelihood we lose by adopting suboptimal assumptions.

To wit, we calculate different likelihood values given the 50,000 Hawkes process realizations we generated for each triggering function in sections 2.3 and 2.4. For each exponential history $H = \{t_i\}_{i=1}^{n}$, we compute the log-likelihood (2.2) of the EM parameters $(\hat{\mu}_{\exp}(H), \hat{\theta}_{\exp}(H))$ and the exponential ground-truth parameters $(0.05, 0.5, 6)$.

Table 3. *Log-likelihood loss vs. maximum*

Model Parameters	Correct EM	Incorrect EM	Correct Ground truth	Incorrect "Ground truth"
Exponential	0	−0.11	−1.51	−7.47
Power-law	0	−0.05	−1.50	−7.66

We also calculate $(\hat{\mu}_{\text{pow}}(H), \hat{\theta}_{\text{pow}}(H))$, the parameters maximizing the likelihood under a power law model, and compute their likelihood. For comparison we also compute the likelihood for the power-law ground-truth parameters $(0.05, 0.5, 3)$. We then repeat the process *mutatis mutandis* for each power-law history. In this way we hope to quantify the loss incurred by using the "wrong" model for the triggering function, as compared to the loss incurred by using the "right" model with the "wrong" parameters. Because both models have the same number of parameters, the penalty term of the Akaike information criterion is unnecessary.

Table 3 summarizes the results. The numbers are the average loss in log-likelihood from the maximum by adopting a certain model and parameters across all realizations. The first column is the loss from using the "correct" model and the EM parameters. As expected this is 0 for both models. The second column is the loss from adopting the "incorrect" model but using the likelihood-maximizing parameters given that model. The third column is the loss from using the "correct" model's ground-truth parameters rather than the likelihood-maximizing parameters. The fourth column is the loss using the "incorrect" model's ground-truth parameters. We have no reason to expect this last category to perform well; we include it for a sense of scaling.

In both cases, the loss from using the EM parameters assuming the wrong model is significantly less than the loss from using the right model with the ground-truth parameters. To emphasize, these are the parameters *that actually generated the histories*, and they still are not as good as a certain set of parameters attached to the wrong model (though not every set, as the fourth column makes clear). The clear moral is that selecting the "correct" model is not as important as finding the likelihood-maximizing parameters once a model has been selected. This justifies the common assumption of the convenient exponential form for the triggering function.

3 The missing data problem

In this section we state the missing data problem and discuss its numerical solution. We take a variational approach, maximizing a discriminant function subject to certain constraints. For the numerics we adapt the curvilinear method of Wen & Yin [30].

3.1 Objective functions

Suppose that we have records of N emails sent among a social network of V members, as in the IkeNet data set. But suppose that for some subset of the emails, we do not know who sent or received them. More generally, we want to identify which of the M edges

each email in the subset was drawn from. Because M scales with V^2, a direct approach enumerating all possibilities and checking them is not scaleable. Instead, we relax the problem as in [27].

Number the M connections from 1 to M. (The order does not matter.) The history of events is $H = \{t_i\}_{i=1}^N$. This history is partitioned into C, the events for which we know which connection the event happened on, and I, the incomplete-information event. The complete set has the obvious partition $C = \bigcup_{m=1}^M C_m$ into the histories associated to each connection.

We present four methods for classifying the incomplete events. The first two are simple, model-free methods based on basic statistics of H. The other two are variational methods maximizing a sort of score function. In each case we have what amounts to a family of discriminant functions, one for each of the M connections. The value of the discriminant function for $t_i \in I$ on connection m is $x_{i,m}$. We speak of x_i as the vector of weights associated to $t_i \in I$. Not every x_i need belong to the same space, or even have the same dimension, as the others. We need define $x_{i,m}$ only for those edges m to which t_i could belong. For example, if we know that one of the parties to an email was cadet 1, we need not consider the weight on the connection between cadets 2 and 3.

The first classification method is a *method of modes*, which sets $x_{i,m} = |C_m|$. The only dependence on i comes from the fact that we do not set $x_{i,m}$ if message i could not have been sent on connection m. The second method is a nearest-neighbor weighting, which weights depending on the proximity in time (forward or backward) of the nearest known event: $x_{i,m} = \max\{|t_i - t_j|^{-1} : t_j \in C_m\}$.[1] These two methods are in a sense dual to one another: the method of modes is a simple, model-free, global method, and the nearest-neighbor method is a simple, model-free, local method. They can serve as a benchmark for the other methods, which assume a Hawkes process model and in so doing incorporate both global and local information.

The third method for $x_{i,m}$ is a relaxed maximum likelihood method. The likelihood of a given history and parameter set is

$$L = \left(\prod_{t_i \in I} \lambda_{m_i}(t_i) \right) \prod_{m=1}^{M} \left(\prod_{t_i \in C_m} \lambda_m(t_i) \right) e^{-\int_0^T \lambda_m(t) dt}.$$

A true MLE approach would find the $\{m_i : t_i \in I\}$ maximizing the likelihood. However, there are $M^{|I|}$ possible values, so this approach quickly becomes infeasible as M and $|I|$ grow. We instead consider a relaxed problem, in which we maximize the related quantity

$$L = \prod_{m=1}^{M} \left(\prod_{t_i \in C_m} \lambda_m(t_i; x) \right) \left(\prod_{t_i \in I} \lambda_m(t_i; x)^{x_{i,m}} \right) e^{-\int_0^T \lambda_m(t;x) dt}$$

where

$$\lambda_m(t; x) = \mu_m + \sum_{t_i \in C_m, t_i < t} g(t - t_i; \theta_m) + \sum_{t_i \in I, t_i < t} x_{i,m} g(t - t_i; \theta_m).$$

If we restrict the vector x_i to be a Kronecker delta, we recover the original maximum likelihood. The relaxation is in the constraint on each x_i: $\|x_i\|_2 = 1$ and $x_{i,m} \geq 0$ for all

[1] The maximand can be replaced with $(\delta + |t_i - t_j|)^{-1}$ if some t_i coincides with some t_j.

Table 4. *Objective functions*

Method	$F(x)$
SSB	$\sum_{m=1}^{M} \sum_{t_i \in I} x_{i,m} \lambda_m(t_i; x)$
MRL	$\sum_{m=1}^{M} \left(\sum_{t_i \in C_m} \log \lambda_m(t_i; x) + \sum_{t_i \in I} x_{i,m} \log \lambda_m(t_i; x) - \sum_{t_i \in I} x_{i,m} G_m(T - t_i) \right)$

m. In practice we will maximize not L directly but a quantity that is off by an additive constant from its logarithm, namely

$$F_{\mathrm{MRL}}(x) = \sum_{m=1}^{M} \left(\sum_{t_i \in C_m} \log \lambda_m(t_i; x) + \sum_{t_i \in I} x_{i,m} \log \lambda_m(t_i; x) - \sum_{t_i \in I} x_{i,m} G_m(T - t_i) \right),$$

where $G_m(t) = \int_0^t g(s; \theta_m) ds$. (MRL here stands for *maximum relaxed likelihood*.)

The fourth method is the Stomakhin–Short–Bertozzi (SSB) method outlined in [27]. This essentially maximizes F_{SSB} defined by

$$F_{\mathrm{SSB}}(x) = \sum_{m=1}^{M} \sum_{t_i \in I} x_{i,m} \lambda_m(t_i; x)$$

subject to similar constraints on each x_i.

3.2 Numerical implementation

Computing x for the method of modes and nearest-neighbor method is straightforward. Constrained maximization of F_{SSB} and F_{MRL} requires more care. Both optimizations have the form

$$\boxed{\max F(x) \text{ s.t } \|x_i\|_2 = 1 \, \forall i \text{ and } x_i, m \geq 0 \, \forall i, m.}$$

The forms of F are summarized in Table 4. This is a variational approach to the classification problem. Variational methods have had success in various applications, including image processing [2, 3].

Though F_{SSB} was created to approximate the behavior of F_{MRL}, the two functions have different properties. For example, F_{SSB} is a quadratic function with all positive coefficients, so within the feasible set all its partial derivatives are positive. This means that every component of the maximizing x is positive. (See the appendix for a proof.) Not so for F_{MRL}:

$$\frac{\partial F_{\mathrm{MRL}}}{\partial x_{i,m}} = \log \lambda_m(t_i; x) + \sum_{t_j \in C_m;\, t_j > t_i} \frac{g_m(t_j - t_i)}{\lambda_m(t_j; x)} + \sum_{t_j \in I;\, t_j > t_i} \frac{x_{i,m} g_m(t_j - t_i)}{\lambda_m(t_j; x)} - G_m(T - t_i).$$

The two sums are positive, but the logarithm need not be, and $-G_m(T - t_i)$ can easily be the dominant term.

We used a modified version of the curvilinear search described in [30]. In this section we introduce that algorithm, discuss our modifications, and finally present the whole algorithm for reference.

3.2.1 *Wen & Yin's curvilinear search*

Gradient ascent is probably the most basic and intuitive iterative method for smooth optimization, but it does not preserve norms. Wen & Yin [30] present a curvilinear adaptation that preserves orthogonal constraints of the form $X^{\mathrm{T}}X = I$, of which our constraint $\|x_i\|_2 = 1$ is a special case. Let $F_{x_i}(x)$ denote the gradient of F with respect to x_i, evaluated at x. At each step, given x_i, Wen & Yin's algorithm yields a new point y_i that lies on the geodesic that is the curvilinear projection of the ray starting at x_i and pointing in the direction of $F_{x_i}(x)$.

Given x and a step size $\tau > 0$, the method computes

$$y_i(\tau, x) = (I + \tfrac{\tau}{2}A)^{-1}(I - \tfrac{\tau}{2}A)x_i,$$

where

$$A = x_i F_{x_i}(x)^{\mathrm{T}} - F_{x_i}(x)x_i^{\mathrm{T}}.$$

A straightforward calculation verifies that if $\|x_i\|_2 = 1$, then $\|y_i(\tau, x)\|_2 = 1$ for all $\tau > 0$. It is likewise straightforward to see that $y_i(\tau, x)$ can be written as

$$y_i(\tau, x) = (1 - \beta_2)x_i + \beta_1 F_{x_i}(x), \tag{3.1}$$

where

$$\begin{aligned}\beta_1 &= \frac{\tau}{1 + (\frac{\tau}{2})^2\delta_i(x))},\\ \beta_2 &= (F_{x_i}(x)^{\mathrm{T}}x_i + \tfrac{\tau}{2}\delta_i(x))\beta_1,\\ \delta_i(x) &= \|F_{x_i}(x)\|_2^2 - (F_{x_i}(x)^{\mathrm{T}}x_i)^2.\end{aligned}$$

(This is Lemma 4 in [30].) Because $\|x_i\|_2 = 1$, the Cauchy-Schwarz inequality ensures that $\delta_i(x) \geq 0$.

3.2.2 *Inequality constraints*

The algorithm in [30] simply sets $x_i^{(k+1)} = y_i(\tau, x_i^{(k)})$, with some adaptive time stepping for τ. While this preserves $\|x_i\|_2$, it does not preserve the signs of the components of x_i. Our inequality constraint $x_{i,m} \geq 0$ forces us to concern ourselves with the signs.

If each component of $x^{(k)}$ (the k^{th} iterate) is positive but some component of $y_i(\tau, x_i^{(k)})$ is negative, then there exists a largest $\sigma \in (0, \tau)$ so that $y_i(\sigma, x^{(k)})$ has all non-negative components. This σ is actually straightforward to compute, because each equation of the form $y_{i,m}(\sigma, x_i^{(k)}) = 0$ is a quadratic equation in σ.

However, we found that this technique was slow in practice because it only allows one dimension of x_i to reach 0 at a time. When $F = F_{\mathrm{SSB}}$, many components of the maximizer x_i^* are close to 0, so we would like to allow many of them to reach 0 at once so they can then turn around and find their correct (small, positive) value. When $F = F_{\mathrm{MRL}}$, many dimensions will ultimately belong to the active set, and we would like to identify several of them at a time if possible. Therefore, we adopted the less elegant but faster method of setting $z = \max(0, y_i(\tau, x_i^{(k)}))$, with the max done componentwise, and then redistributing the mass to preserve the ℓ^2 norm, i.e. $\tilde{x}_i^{(k+1)} = z/\|z\|_2$.

If we adopt $x_i^{(k+1)} = \tilde{x}_i^{(k+1)}$, then it may have components that are zero and that

will become negative after another iteration of the curvilinear search. If we continue with these components, the algorithm may hang because the projection back to the sphere may become parallel to the curvilinear search direction. We can prevent this if we acknowledge that any dimensions m for which $y_{i,m}(\tau, \tilde{x}_i^{(k+1)}) < 0$ belong to the active set of inequality constraints. Noting from (3.1) that $y_{i,m}(\tau, x)$ and $F_{x_i}(x)$ have the same sign when $x_{i,m} = 0$, we set $x_i^{(k+1)} = P(x, \tilde{x}_i^{(k+1)})\tilde{x}_i^{(k+1)}$, where $P(x, \tilde{x}_i^{(k+1)})$ is the projection onto the subspace of those dimensions m for which $\tilde{x}_{i,m}^{(k+1)} > 0$ or $F_{x_i} > 0$, with the derivative evaluated at x except with x_i replaced with $\tilde{x}_i^{(k+1)}$. (As we iterate, we also remove dimensions from F and ∇F so that dot products with x_i still make sense and so that we are not calculating derivatives unnecessarily.)

When $F = F_{\mathrm{SSB}}$ the solution can have many small positive components. It is possible that at $x_i^{(1)}$ many components $x_{i,m}^{(1)}$ are small and positive but have $y_{i,m}(\tau, x^{(1)}) < 0$, and many others are zero but have $y_{i,m}(\tau, x^{(1)}) > 0$. These sets of components trade places in $x_i^{(2)}$, and the next iteration will send it back to very close to $x_i^{(1)}$. If enough components keep "trading places" like this it can cause the algorithm to hang without reaching the stopping criterion. We found that when $|I|$ was large this happened a small but nontrivial percentage of the time. We also found that we could eliminate the problem by checking the signs of the components of x_i versus $y_i(\tau, x)$. If most were different, we tried $y_i(\tau/2, x)$, and then $y_i(\tau/4, x)$, and so on until a majority of the signs were preserved.

Once the iteration completes, we need to check that the dimensions we have projected away still correspond to active constraints. If they do not, we project $x^{(k)}$ into a larger space including the inactivated dimensions and resume iterating.

3.2.3 *Stopping criterion*

Wen & Yin [30] give a stopping criterion of $\|\nabla F\|_2 < \epsilon$. Our stopping criterion must be different, because we do not expect $\|\nabla F\|_2$ to decrease to 0 as we iterate. (Indeed, as noted above, the components of ∇F_{SSB} are always positive.) Instead we look for ∇F to be normal to the constraint surface. Since the constraint surface is a sphere, this means we want $\nabla F \cdot x$ to be large relative to the size of ∇F. Specifically, our stopping criterion is

$$\min_{t_i \in I} \frac{|F_{x_i}(x_i^{(k)}) \cdot x_i^{(k)}|}{\|F_{x_i}(x_i^{(k)})\|_2} > 1 - \epsilon.$$

The absolute value in the numerator is necessary only if every $F_{x_i}(x_i^{(k)})$ is negative. This can happen when $F = F_{\mathrm{MRL}}$ but not when $F = F_{\mathrm{SSB}}$.

3.2.4 *Algorithm*

while $\max_{t_i \in I} |F_{x_i}(x_i) \cdot x_i| / \|F_{x_i}(x_i)\|_2 > \epsilon$ **do**
  for $i = 1 : |I|$ **do**
    $v = F_{x_i}(x)$
    $\delta = \|v\|_2^2 - (v^{\mathrm{T}} x_i)^2$
    $\beta_1 = \tau / (1 + (\frac{\tau}{2})^2 \delta)$
    $\beta_2 = (v^{\mathrm{T}} x_i + \frac{\tau}{2}\delta)\beta_1$

$y = (1 - \beta_2)x_i + \beta_1 F_{x_i}(x)$
$\overline{\tau} = \tau$
while most components of y have different signs than x_i **do**
$\overline{\tau} = \overline{\tau}/2$
$\beta_1 = \overline{\tau}/(1 + (\frac{\overline{\tau}}{2})^2\delta)$
$\beta_2 = (v^{\mathrm{T}}x_i + \frac{\overline{\tau}}{2}\delta)\beta_1$
$y = (1 - \beta_2)x_i + \beta_1 F_{x_i}(x)$
end while
$z = \max(0, y)$ componentwise
$\tilde{x} = x$
$\tilde{x}_i = z/\|z\|_2$
$v = F_{x_i}(\tilde{x})$
Let P project the space of x_i to the subspace where $\tilde{x}_{i,m} > 0$ or $v_m > 0$
$x_i = P\tilde{x}_i$
$F_{x_i} = PF_{x_i}$
end for
end while
for $i = 1 : |I|$ **do**
Let Q project the space of x_i into its original, full space
$w_i = Qx_i$
$F_{x_i} = QF_{x_i}$
end for
startover = false
for $i = 1 : |I|$ **do**
$v = F_{x_i}(w)$
for all m in the space of w_i **do**
if m is not in the space of x_i and $v_i > 0$ **then**
Project x_i into its own space augmented with dimension m.
startover = true
end if
end for
end for
if startover **then**
for $i = 1 : |I|$ **do**
Project F_{x_i} into the space of x_i
end for
Return to the start.
end if

3.2.5 *Practical computing considerations*

The most computationally expensive part of our C++ implementation of the algorithm is the computation of the derivative F_{x_i}. Care must be taken to minimize this expense.

For reference, its components for our two choices of F are

$$\frac{\partial F_{\text{SSB}}}{\partial x_{i,m}} = \mu_m + \sum_{t_j \in C_m} g_m(|t_i - t_j|) + \sum_{t_j \in I;\, t_j \neq t_i} x_{j,m} g_m(|t_i - t_j|), \tag{3.2}$$

$$\frac{\partial F_{\text{MRL}}}{\partial x_{i,m}} = \log \lambda_m(t_i; x) + \sum_{t_j \in C_m;\, t_j > t_i} \frac{g_m(t_j - t_i)}{\lambda_m(t_j; x)} + \sum_{t_j \in I;\, t_j > t_i} \frac{x_{i,m} g_m(t_j - t_i)}{\lambda_m(t_j; x)} - G_m(T - t_i).$$

Values of g_m should never be computed "on the fly"; each should be precomputed and stored. Most of these values will be so small that treating them as zero will have a *de minimis* impact on the results, but avoiding computing them (and computing with them) saves tremendous time. Set a small threshold $\eta > 0$, and compute $g_m(t_i - t_j)$ only if it will exceed $\eta\mu_m/|C_m|$, i.e. if $|t_i - t_j| < g_m^{-1}(\eta\mu_m/|C_m|)$. This adds a layer of dependency tracking, but the savings in floating point operations are well worth it.

When $F = F_{\text{SSB}}$, the update formula

$$\frac{\partial F_{\text{SSB}}}{\partial x_{i,m}}(x^{(1)}) = \frac{\partial F_{\text{SSB}}}{\partial x_{i,m}}(x^{(0)}) + \sum_{t_j \in I;\, t_j \neq t_i} g_m(|t_i - t_j|)(x_{j,m}^{(1)} - x_{j,m}^{(0)})$$

can save time when recomputing F_{x_i}. When $F = F_{\text{MRL}}$, a corresponding update formula applies for $\lambda_m(t_j; x)$. The λ values should be tracked, while the logarithm should be computed only when it is needed.

4 Results

Here we present results for different configurations of missing data. First we present results from the IkeNet data set. Then we test the methods on simulated time series on artificial social networks, including some toy networks and some meant to resemble IkeNet. We conclude the section by discussing the results in detail.

In each of our tests we begin with a complete data set, whether it is real (IkeNet) or simulated. Then we knock out some of the information to see whether we can recover it from the rest of the corpus. The information might be a particular email's sender or receiver, an email's sender *and* receiver, or the senders and receivers of several emails. When deleting one record at a time we repeat this for each record in the corpus. When deleting more than one record, exhausting the space of combinations is infeasible, so we take a Monte Carlo approach.

We consider a data recovery method successful when the correct component $x_{i,m}$ has a high weight relative to other components. In particular, we want $x_{i,m}$ to be the greatest component, or perhaps the second or third greatest. This metric was considered previously in [27] based on input from the LAPD. (The context there was solving gang crimes, where narrowing down the list of suspect gangs to three can help detectives.) We also present the results for top 5 and top 10 to showcase a property of the MRL optimizer.

We estimate the Hawkes process parameters using the techniques described in section 2. The SSB and MRL iterations are seeded with the solution from the nearest-neighbor method.

Table 5. *IkeNet: Predictive power for missing sender by method ($|I| = 1$)*

Method	Top 1	Top 2	Top 3	Top 5	Top 10
Modes	27.8%	41.1%	50.0%	62.9%	82.0%
NN	62.9%	75.1%	79.8%	85.3%	92.6%
SSB	63.1%	74.7%	80.0%	85.8%	93.3%
MRL	61.1%	70.0%	72.4%	73.3%	73.6%

Table 6. *IkeNet: Predictive power for missing receiver by method ($|I| = 1$)*

Method	Top 1	Top 2	Top 3	Top 5	Top 10
Modes	30.4%	43.5%	52.1%	64.4%	82.7%
NN	58.0%	73.3%	80.1%	86.6%	93.9%
SSB	59.2%	73.9%	80.6%	87.1%	93.7%
MRL	58.9%	69.0%	71.7%	72.6%	72.8%

4.1 IkeNet

4.1.1 *Unidirectional identity loss, one at a time*

First we took each email in the corpus and saw whether we could determine who sent it knowing its receiver and the rest of the corpus. Repeating this for each email in the corpus meant 8,896 separate runs with $|I| = 1$ each time. The average performance is shown in Table 5.

Table 5 shows that SSB, nearest-neighbor (NN), and MRL guess the correct sender about 60% of the time. There is a clear ranking among them, with SSB outperforming nearest-neighbor and nearest-neighbor outperforming MRL. MRL's relative performance decreases left to right. The method of modes performs poorer than the other methods.

Table 6 shows the results when we repeat the process but try to guess the receiver knowing the sender. The numbers are slightly different, but the same patterns prevail.

4.1.2 *Unidirectional identity loss, missing proportions*

We now consider what happens when larger blocks of data are missing, which will be the case in applications. We selected a percentage of the emails at random and removed the sender or receiver information (chosen randomly for each email). We then attempted to recover the missing data. We repeated this process for 10,000 Monte Carlo runs at each missing percentage.

Table 7 shows the results. As expected, the performance decreases as the missing proportion increases from 5% to 20%, but only by a few percentage points. This demonstrates the methods' robustness to larger missing blocks of data. The method of modes experiences no degradation. This is not a surprise; it returns the same top pairs shown in Table 1 until enough data is missing in the right places that the order statistics change.

Table 7. *IkeNet: Predictive power for unidirectional identity loss ($|I| > 1$)*

$\|I\|/N$	Method	Top 1	Top 2	Top 3	Top 5	Top 10
5%	Modes	29.1%	42.2%	50.9%	63.1%	82.1%
	NN	59.9%	73.5%	79.3%	85.4%	93.0%
	SSB	59.9%	73.5%	79.7%	86.0%	93.3%
	MRL	59.4%	68.9%	71.4%	72.2%	72.4%
10%	Modes	29.1%	42.2%	50.9%	63.1%	82.1%
	NN	59.3%	72.8%	78.6%	84.7%	92.6%
	SSB	58.8%	72.7%	79.0%	85.5%	93.1%
	MRL	58.9%	68.3%	70.7%	71.5%	71.7%
15%	Modes	29.1%	42.1%	50.9%	63.1%	82.1%
	NN	58.7%	72.1%	77.8%	84.1%	92.3%
	SSB	57.7%	71.9%	78.4%	85.1%	92.9%
	MRL	58.3%	67.6%	69.9%	70.7%	70.8%
20%	Modes	29.0%	42.1%	50.8%	63.1%	82.1%
	NN	58.1%	71.4%	77.1%	83.4%	92.0%
	SSB	56.5%	71.1%	77.7%	84.4%	92.5%
	MRL	56.8%	67.0%	69.2%	69.8%	70.0%

Table 8. *IkeNet: Predictive power for bidirectional identity loss ($|I| = 1$)*

Method	Top 1	Top 2	Top 3	Top 5	Top 10
Modes	11.7%	17.5%	20.9%	27.3%	36.7%
NN	37.9%	51.3%	58.5%	65.6%	73.2%
SSB	39.6%	51.1%	57.6%	65.3%	73.0%
MRL	36.4%	47.8%	55.0%	61.4%	66.1%

4.1.3 *Bidirectional identity loss, one at a time*

We repeated the one-at-a-time procedure with deleting both sender and receiver from each email, resulting in *bidirectional identity loss*. Table 8 presents the results. The methods do not perform as well as when only the sender or receiver is missing because instead of choosing among the 22 edges connected to each nodes they must choose among the 253 edges in the complete graph.[2] Nonetheless the local methods guessed the correct edge about 40% of the time and got in the top 3 about 55-60% of the time. MRL still underperforms, but by less than with unidirectional loss. The method of modes continues to underperform all other methods.

Table 9 presents average numerical values of F_{SSB} and F_{MRL} evaluated at the bidirectional identity loss solutions in Table 8.[3] Horizontal comparison of the values is meaning-

[2] Actually there are only 250 edges; as noted above, three pairs of agents exchanged no emails.
[3] The values shown are actually of $F(x) - F_{\min}$ to highlight the differences in scale.

Table 9. *IkeNet: Average energy values for bidirectional identity loss ($|I| = 1$)*

Method	F_{SSB}	F_{MRL}
Modes	45.82	85.62
NN	122.39	99.37
SSB	141.39	99.47
MRL	118.09	101.01

Table 10. *IkeNet: Predictive power for bidirectional identity loss ($|I| > 1$)*

$\lvert I\rvert/N$	Method	Top 1	Top 2	Top 3	Top 5	Top 10
5%	Modes	11.7%	17.5%	20.8%	27.4%	36.8%
	NN	37.5%	50.8%	57.9%	64.9%	72.5%
	SSB	38.6%	50.4%	57.0%	64.4%	72.3%
	MRL	36.1%	47.5%	54.5%	60.8%	65.2%
10%	Modes	11.7%	17.5%	20.8%	27.3%	36.7%
	NN	37.3%	50.3%	57.2%	64.0%	71.4%
	SSB	37.5%	49.3%	55.8%	63.2%	71.2%
	MRL	35.7%	47.0%	53.7%	60.2%	64.3%

Table 11. *IkeNet: Average energy values for bidirectional identity loss ($|I|/N = 5\%$)*

Method	$F_{\mathrm{SSB}}/\lvert I\rvert$	$F_{\mathrm{MRL}}/\lvert I\rvert$
Modes	48.98	83.91
NN	120.77	97.72
SSB	147.17	97.72
MRL	112.56	99.91

less, but vertical comparison is not. The results verify that the SSB and MRL solutions maximize F_{SSB} and F_{MRL}, respectively.

4.1.4 *Bidirectional identity loss, missing proportions*

Table 10 shows the results of the Monte Carlo approach for larger blocks of missing bidirectional data. Bidirectional is much more intensive computationally than unidirectional, so we present proportions only up to 10% here. The degradation is again modest (compare with Table 8), and the ranking of methods is consistent.

Table 11 shows average energy values, normalized by the size of the missing block for comparison with Table 9. The values are close, and the same hierarchies are apparent.

4.2 Simulated time series

We simulate Hawkes processes on two classes of networks. First we consider some toy networks with simple structures. Then we simulate a faux IkeNet (*FauxNet*) using the IkeNet parameters.

4.2.1 *Toy networks*

We use three different configurations of toy networks. Like IkeNet they have 22 nodes, but a known interaction structure. We assume that g is exponential with $\alpha = 0.5$, $\omega = 6$, with the background rate μ varying to show different levels of interaction.

- *Dense*: All nodes are connected to each other (a complete graph), with a low rate of interaction ($\mu = 0.03$).
- *Sparse*: The nodes are arranged in a ring. Each node is connected to its two neighbors and to the node opposite it in the ring, so that the graph looks like a wheel with spokes (but there is no node at the axle). Interaction rates between connected nodes are high ($\mu = 0.1$). Unconnected nodes do not interact.
- *Pseudosparse*: A complete graph, with high interaction ($\mu = 0.1$) between the nodes connected in the sparse graph and low interaction ($\mu = 0.03$) between other pairs.

Table 12 presents the results for Monte Carlo simulation. For each network, we adopted bidirectional identity loss for each record in succession, and then averaged the results over each Monte Carlo simulation. Table 12 compares with Table 8.

The method of modes performs very poorly here compared with IkeNet, because the toy networks lack the handful of highly active connections evident in Table 1. NN, SSB, and MRL perform similarly, as with IkeNet, but here MRL outperforms NN. SSB still outperforms them both. Unsurprisingly, all methods perform better on the sparse network than on the dense network, but the local methods perform very well compared to the method of modes even on the dense network. Interestingly, though the performance of the method of modes on the pseudosparse network is between its performances on the dense and sparse networks, the local methods perform worst on the pseudosparse network. This is because the local methods perform poorer as the number of connections experiencing a burst of activity at any given time increases. This strength of this effect decreases as we move from top 1 to top 10, and indeed this is reflected in Table 12.

4.2.2 *FauxNet*

As with the toy networks, we took a Monte Carlo approach to FauxNet, the simulated IkeNet, and present results for bidirectional identity loss in Tables 13 and 14. The method of modes performs almost the same as in IkeNet (see Tables 8 and 10). The other methods perform better here by several percentage points.

4.3 Discussion

In all our results, the local methods (nearest-neighbor, SSB, and MRL) strongly outperform the purely global method of modes. This suggests that most of the information in

Table 12. *Toy networks: Predictive power for bidirectional identity loss ($|I| = 1$)*

Network	Method	Top 1	Top 2	Top 3	Top 5	Top 10
Dense	Modes	1.0%	1.9%	2.7%	4.3%	7.9%
	NN	21.4%	36.5%	47.0%	59.3%	69.0%
	SSB	27.4%	41.6%	50.6%	61.0%	69.7%
	MRL	26.4%	40.9%	49.6%	57.9%	61.9%
Sparse	Modes	4.5%	8.6%	12.4%	20.1%	37.7%
	NN	36.9%	55.5%	65.0%	72.6%	78.8%
	SSB	40.8%	57.5%	65.8%	73.0%	79.6%
	MRL	39.8%	55.9%	62.0%	63.6%	64.9%
Pseudosparse	Modes	1.5%	2.8%	4.2%	6.7%	12.5%
	NN	17.9%	31.4%	41.5%	54.7%	67.6%
	SSB	23.7%	36.8%	45.8%	57.0%	68.3%
	MRL	23.0%	36.2%	45.1%	54.9%	61.5%

Table 13. *FauxNet: Predictive power for bidirectional identity loss ($|I| = 1$)*

Method	Top 1	Top 2	Top 3	Top 5	Top 10
Modes	11.6%	17.4%	21.0%	27.2%	36.7%
NN	49.4%	60.2%	63.9%	66.9%	70.3%
SSB	53.6%	63.2%	66.8%	70.1%	74.3%
MRL	48.5%	60.6%	64.5%	65.9%	66.0%

Table 14. *FauxNet: Predictive power for bidirectional identity loss ($|I|/N = 5\%$)*

Method	Top 1	Top 2	Top 3	Top 5	Top 10
Modes	11.6%	17.4%	21.0%	27.2%	36.7%
NN	49.0%	59.6%	63.3%	66.2%	69.6%
SSB	52.7%	62.2%	65.9%	69.3%	73.6%
MRL	48.1%	59.9%	63.7%	65.2%	65.3%

these sorts of records is local. Meanwhile, with IkeNet the model-free nearest-neighbor method performs comparably to the variational methods (SSB and MRL) developed in section 3. With the simulated Hawkes process data it underperforms SSB and, in some places, MRL, but not by nearly the margin that the method of modes does. This suggests that the Hawkes process is an imperfect model for real human communication like the IkeNet data, but the loss in assuming it is modest. On the other hand, the loss in assuming no model at all (i.e. using nearest-neighbor) is also modest and has the virtue of being simpler to implement, understand, and communicate outside technical literature.

The improvement in MRL's performance as it moves from top 5 to top 10 is consider-

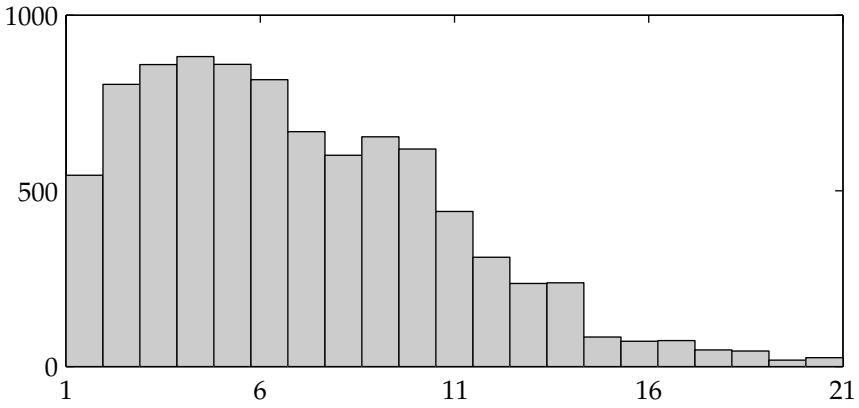

FIGURE 6. Histogram of $\|x_{\mathrm{MRL}}\|_0$ for bidirectional identity loss, $|I| = 1$, for all 8,896 cases.

ably lower than it is for the other methods. Figure 6 reveals why. It shows a histogram of $\|x_{\mathrm{MRL}}\|_0$, the number of nonzero components of x_{MRL}, for each bidirectional $|I| = 1$ case. The median is 6, and $\|x_{\mathrm{MRL}}\|_0 \leq 5$ in about 44% of cases. In these cases, if the correct pair is not in the top 5 then it will not be in the top 10, either. SSB, by contrast, always has full ℓ^0 norm (see the appendix for a proof), and even if the correct pair has only a small positive weight it is often larger enough than the other small positive weights to make it to the top 10. Of course, MRL has even fewer positive components in the unidirectional case, explaining why it underperforms less in bidirectional identity loss Thus SSB's density is capturing some faint information that MRL misses by being so sparse. If a likelihood approach like MRL is to beat SSB it will likely have to mimic this ability.

All the methods except the method of modes perform better on FauxNet than on IkeNet. Furthermore, SSB and MRL perform better relative to nearest-neighbor on the simulated time series than they do on IkeNet data. Both these observations suggest that the Hawkes process is an imperfect model for the behavior driving IkeNet.

5 Conclusion

We demonstrated that, when estimating the parameters of a Hawkes process from data, choosing a parameterization for the triggering function is less important than using the correct values of the parameters. We then developed a method for filling in missing data for interactions within social networks and presented some results from the IkeNet data set. The method's power even when the proportion of missing data increases has implications for security, surveillance, and privacy. In particular, it suggests that access to even a fraction of a complete record can reveal a great deal of information about the remainder, emphasizing the need for robust access controls.

Future work should address how network structure impacts the ability to fill in missing data. Exogenous information (for example, the leadership relationships among the IkeNet cadets) may also be able to boost the method's power. Future work might also seek an objective function combining MRL's fidelity to the original likelihood with SSB's solution

density. However, as noted, modeling IkeNet's email behaviors with Hawkes processes has its limits, so consideration of other classes of self-exciting point processes may be warranted.

Acknowledgements

This research was supported by NSF grant DMS-0968309, AFOSR MURI grant FA 9550-10-1-0569, ONR grant N000141210838, and ARO MURI grant W911NF-11-1-0332. The first author is supported by a National Defense Science and Engineering Graduate (NDSEG) Fellowship. We thank Eric Fox, Martin Short, Alexey Stomakhin, and Wotao Yin for helpful discussions.

Appendix

We prove that the SSB weight vector always has all positive components, as a corollary of the following.

Proposition *Let $n \geq 2$, and let D be the portion of the unit sphere in the non-negative orthant of R^n, i.e. $D = \{x \in \mathsf{R}^n : \|x\|_2 = 1, x_i \geq 0\, \forall i\}$. Let $f : \mathsf{R}^n \to \mathsf{R}$ be differentiable with all positive partial derivatives on the non-negative orthant. Then there exists $x^* \in D$ maximizing f on D, and $\|x^*\|_0 = n$.*

Proof x^* exists because f is continuous and D is compact. Suppose by way of contradiction that $\|x^*\|_0 < n$. Without loss of generality, $x_1^* = 0$. By assumption $\|x^*\|_2 = 1$, so without loss of generality $x_2^* > 0$. Define $\xi : [0, x_2^*] \to \mathsf{R}^n$ by

$$\xi_i(t) = \begin{cases} t & \text{if } i = 1, \\ \sqrt{(x_2^*)^2 - t^2} & \text{if } i = 2, \\ x_i^* & \text{if } 3 \leq i \leq n. \end{cases}$$

Then $\xi(t) \in D$ for every t. Because f is differentiable there exist $t_0 > 0$ and $h : (0, t_0) \to \mathsf{R}$ such that $h(t) = o(t)$ as $t \to 0$, and if $0 < t < t_0$ then

$$f(\xi(t)) = f(x^*) + t\nabla f(x^*)^{\mathrm{T}} \xi'(0) + h(t).$$

Easy computations show that $\xi_1'(0) = 1$, $\xi_2'(0) = 0$, and $\xi_i'(0) = 0$ if $3 \leq i \leq n$, so

$$f(\xi(t)) = f(x^*) + t\frac{\partial f}{\partial x_1}(x^*) + h(t).$$

By assumption $\frac{\partial f}{\partial x_1}(x^*) > 0$, so there exists $t_1 \in (0, t_0]$ such that if $0 < t < t_1$ then $|h(t)|/t < \frac{1}{2}\frac{\partial f}{\partial x_1}(x^*)$, in which case

$$f(\xi(t)) > f(x^*) + t\frac{\partial f}{\partial x_1}(x^*) - \frac{t}{2}\frac{\partial f}{\partial x_1}(x^*) > f(x^*),$$

contradicting the assumption that x^* maximizes f on D. Thus in fact $\|x^*\|_0 = n$. □

The assumptions that all partial derivatives of f on the non-negative orthant be positive was stronger than necessary; it would have sufficed that, if $y \in D$ has a zero component $y_i = 0$, then $\frac{\partial f}{\partial x_i}(y) > 0$. Nonetheless it is clear from (3.2) that F_{SSB} satisfies the stronger assumption except in the trivial degenerative case when some $\mu_m = 0$.

References

[1] Barabási, A.-L. 2005 The origin of bursts and heavy tails in human dynamics. *Nature* **435**: 207–11.

[2] Chambolle, A., Caselles, V., Cremers, D., Novaga, M. & Pock, T. 2010 An introduction to total variation for image analysis. In Fornasier, Massimo, ed. 2010 *Theoretical foundations and numerical methods for sparse recovery*. Berlin. De Gruyter: 263–340.

[3] Chan, T. F. & Shen, J. 2005 *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. Philadelphia. SIAM.

[4] Cho, Y. S., Galstyan, A., Brantingham, P. J. & Tita, G. 2014 Latent self-exciting point process model for spatial-temporal networks. *Discrete and Continuous Dynamical Systems B* **19**: 1335–54.

[5] Crane, R. & Sornette, D. 2008 Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences* **105**: 15649–53.

[6] Csermely, P., London, A., Wu, L.-Y., & Uzzi, B. 2013 Structure and dynamics of core/periphery networks. *Journal of Complex Networks* **1**: 93–123.

[7] Egesdal, M., Fathauer, C., Louie, K., Neuman, J., Mohler, G. & Lewis, E. 2010 Statistical modeling of gang violence in Los Angeles. *SIAM Undergraduate Research*.

[8] Fox, E. W., Short, M. B., Schoenberg, F. P., Coronges, K. D. & Bertozzi, A. L. 2014 Modeling e-mail networks and inferring leadership using self-exciting point processes. Submitted to *Journal of the Americal Statistical Association*.

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

[10] —. 1971 Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society B* **33**: 438–43.

[11] Hegemann, R. A., Lewis, E. A., & Bertozzi, A. L. 2013 An "Estimate & Score Algorithm" for simultaneous parameter estimation and reconstruction of incomplete data on social networks. *Security Informatics* **2**: 1.

[12] Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.-F. & Van den Broeck, W. 2011 What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of Theoretical Biology* **271**: 166–80.

[13] Lewis, E., & Mohler, G. 2011 A nonparametric EM algorithm for multiscale Hawkes processes. Preprint.

[14] Lewis, E., Mohler, G., Brantingham, P. J., & Bertozzi, A. 2010 Self-exciting point process models of insurgency in Iraq. UCLA CAM Report 10-38.

[15] Lewis, P. A. W. & Shedler, G. S. 1979 Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly* **26**: 403–13.

[16] Marsan, D. & Lengliné, O. 2008 Extending earthquakes' reach through cascading. *Science* **319**: 1076–79.

[17] Masuda, N., Takaguchi, T., Sato, N. & Yano, K. 2013 Self-exciting point process modeling of conversation event sequences. In Holme, P. & Saramäki, J., eds. 2013 *Temporal networks*. Berlin. Springer-Verlag: 245–64.

[18] Miritello, G., Moro, E., & Lara, R. 2011 Dynamical strength of social ties in information spreading. *Physical Review E* **83**: 045102(R).

[19] Mohler, G. 2013 Modeling and estimation of multi-source clustering in crime and security data. *Annals of Applied Statistics* **7**: 1525–39.

[20] Ogata, Y. 1981 On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory* **27**: 23–31.

[21] — 1998 Space-time point process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics* **50**: 379–402.

[22] — 1999 Seismicity analysis through point-process modeling: a review. *Pure and Applied Geophysics* **155**: 471–501.

[23] Ozaki, T. 1979 Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics* **31**: 145–55.

[24] Paxson, V. & Floyd, S. 1995 Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking* **3**: 226–44.

[25] Rubin, I. 1972 Regular point processes and their detection. *IEEE Transactions on Information Theory* **18**: 547–57.

[26] Rybski, D., Buldyrev, S. V., Havlin, S., Liljeros, F., & Makse, H. A. 2009 Scaling laws of human interaction activity. *Proceedings of the National Academy of Sciences* **106**: 12640–45.

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

[28] Vázquez, A., Oliveira, J. G., Dezsö, Z., Goh, K.-I., Kondor, I., & Barabási, A.-L. 2006 Modeling bursts and heavy tails in human dynamics. *Physical Review E* **73**: 036127.

[29] Veen, A. & 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**: 614–24.

[30] Wen, Z. & Yin, W. 2013 A feasible method for optimization with orthogonality constraints. *Mathematical Programming A* **142**: 397–434.