Parameter Estimation for Persistent Communication Cascades

6.867 Final Project

Steve Morse & Phil Chodrow , December 13, 2016

1 Introduction

A wide variety of processes in human social networks may be understood as spreading processes, in which resources, disease, or information proliferates through pairwise connections between individuals. Models of these processes have therefore received attention in fields ranging from sociology to marketing to statistical physics. A standard assumption of such models is that the rate and kind of pairwise interactions depend only on the current state, not the larger process history. This Markov-like assumption leads to substantial analytic simplification [14], and also acknowledges the difficulty of obtaining data at adequate granularity to model non-Markovian processes [11]. However, the growing availability of massive, passive data has allowed more fine-scaled research into spreading processes, and recent results have shown that many important processes are dramatically non-Markovian (see [1, 7, 12] among others).

One promising route for modeling non-Markovian network spreading processes interprets edge activity as arrivals in a multi-dimensional, mutually exciting point process, the so-called *Hawkes process* [2]. This model has been found to be especially useful in the modeling of network communication processes, where the non-Markovian structure has a natural interpretation: communication events cluster in conversations, rather than arriving at constant rate ([6], [16].) In previous work [8], S. Morse presented a method for extracting persistent group conversation structure from communication data when content is not known (specifically, cell phone record metadata).

In this article, we develop and apply methods for fitting Hawkes process models to persistent communication cascades, thereby quantifying their non-Markovian, self-exciting structure. We demonstrate that a non-homogeneous point process better models the data than homogeneous ones, and present a method for parameter estimation of these non-homogeneous models using a regularized MAP EM scheme with validation. This regularization method is, to our knowledge, novel. Our results lay foundation for greater physical understanding of these processes, as well as future work estimating intra-cascade structure.

2 Data

2.1 Unprocessed data

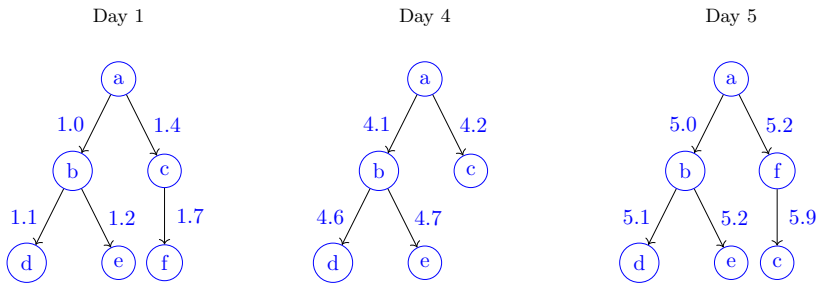
Our data consists of mobile phone Call Detail Records (CDRs) for one mid-size European city over a 13-month period. On average, there are approximately 280k (280×10^3) unique users per month, who contribute to a total 5.8 million call/SMS events. There is a two-month gap in the data for which no records are available. A single call event in the data consists simply of the caller, callee, time stamp, and duration. This is proprietary data available to both authors through research grants.

2.2 Persistent cascades

Because we have access only to metadata, we cannot directly infer the content or importance of any given call event. But suppose that we observe, in some short period, user a call users b and c, who call users e and f, and then we see this pattern or repeated every few days over many weeks or months. Such structure defines a *persistent cascade* of information diffusion in this communication network, and may therefore correspond to important conversatino content.

We give a more detailed exposition of this concept, and our method of identifying and extracting these structures, in the Appendix. This is based on previous work by S. Morse [8]. As a short example, consider the following group of similar observed cascading communication structures, with time stamps on edges on a day

scale:



and the corresponding sequence of events:

$$\{1.0, 1.1, 1.2, 1.3, 1.4, 1.7, 4.1, ..., 5.2, 5.9\}.$$

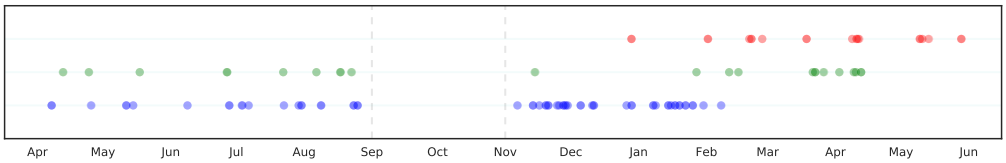


Figure 1: Three example sequences from the data resulting from the persistence analysis outlined in the Appendix. Dots represent call events within a persistent cascade, and so are calls between approximately the same users, in approximately the same order. There is remarkable consistency on the scale of months to a year. The dashed lines show the 2-month period of missing data that we will use to split training/validation.

In this article, we build upon previous work by taking the persistent cascade structure and associated call sequence as given. That is, we use the sequences of call events within these already identified persistent group conversations as a starting point, and we focus on modeling, predicting, and analyzing them. Our (processed) data therefore consists of $\mathcal{D} = \{\tau^{(i)}\}$ where $\tau^{(i)} = \{t_1^{(i)}, ..., t_{n_i}^{(i)}\}$ is sequence of time stamps corresponding to the sequence of call events in the ith group conversation.

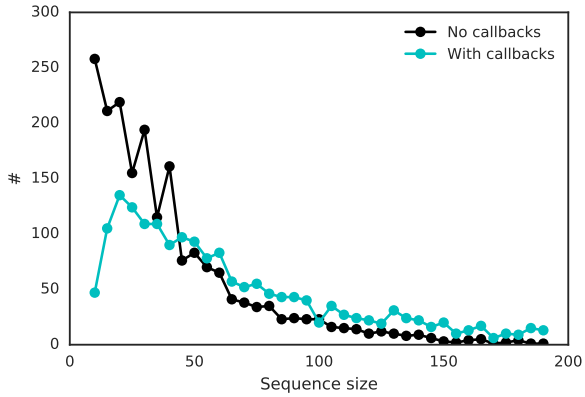


Figure 2: Distribution of sequence lengths, both with and without "callback" events (i.e. subsequent calls between users within the cascade, during the time interval of the cascade).

Some examples of sequences $\tau^{(i)}$ are shown in Figure 1. We note remarkable consistency on the scale of months to a year. We see interesting stories developing: in the first sequence, a new group appears to form (possibly new friends from the holidays?); in the third sequence, there is a crescendo of activity followed by the group completely vanishing (possibly planning an event?). We also note the 2-month break in the data — we do not have observations during this period, and will use this as a convenient way to separate our training and validation data.

Finally, Fig. 2 shows the distribution of number of calls by sequence size for both the standard sequences as outlined above and in the appendix, and when we include all repeat, or callback, calls that also occur within the time interval of an identified persistent structure (which would normally be discarded). This allows us to increase the size of our sample without too much disrupting the already identified causal nature of the sequence.

3 Methods

3.1 Hawkes Process

A *counting process* is a stochastic process $N(t) : t \geq 0$ taking values in $\mathbb{N}$, and $N(0) = 0$, is almost surely finite, and is a right-continuous step function with intervals $+1$. We refer to each step as an *arrival*. We also denote with $H(u) = \{t_i\}_{t=1}^{u}$ the *history* of arrivals up to a time u.

If a sequence of random variables $T = (T_1, T_2, ...)$ taking values in $[0, \infty)$ has $P(0 \leq T_1 \leq T_2 \leq ...) = 1$, and the number of points in a bounded region is a.s. finite, then T is called a *(simple) point process*. The conditional intensity function $\lambda(t)$ of the process N is

$$\lambda(t_0) = \lim_{\delta t \to 0} \frac{\mathbb{E}[N(t_0 + \delta t) - N(t_0)|H(t_0)]}{\delta t} \tag{1}$$

$$= \left. \frac{\partial \mathbb{E}[N(t)|H(t_0)]}{\partial t} \right|_{t=t_0} . \tag{2}$$

The conditional intensity is naturally understood as the infinitesimal arrival rate in the process N.

Definition 3.1 (Hawkes process)**.** A counting process N is a *Hawkes process* (after [2]) if the conditional intensity function has the parameterized form

$$\lambda(t; \Theta) = \mu + \int_0^t g(t - u; \theta) \mathrm{d}N(u) = \mu + \sum_{t_i < t} g(t - t_i; \theta) \ , \tag{3}$$

where $\Theta = \{\mu, \theta\}$, μ is the *background intensity* $\mu > 0$ and θ the parameters of a *triggering function* $g : \mathbb{R}^+ \to \mathbb{R}^+$ (also sometimes called *excitation function*).

Note that when $g \equiv 0$, we recover the (homogeneous) *Poisson process* with rate μ. In this case the intensity is independent of the history H, reflecting another facet of the memorylessness property of the Poisson process. In contrast, a Hawkes process with $g > 0$ is *self-exciting*: recent arrivals increase the value of the intensity function, thereby generating more arrivals. This property results in stronger "clustering" of arrival events than observed in homogeneous Poisson processes.

A common choice of triggering function is a scaled exponential function

$$g(t) \triangleq \alpha \omega e^{-\omega t}. \tag{4}$$

This has an intuitive form if we interpret the HP as a branching process. That is, when the intensity $\lambda(t) = \mu$, then any arrival is called an *immigrant* or a *parent* event, and any immediately subsequent event (where now $\lambda(t) > \mu$ due to the excitation of $g(\cdot)$) is an *offspring*. Now we can interpret $\omega > 0$ as controlling the rate of decaying influence from previous events, and $\alpha > 0$ controlling the *branching ratio*, or likelihood of an arrival causing another arrival.

We note that, when $\alpha > 1$, the process N is nonstationary; i.e. $\mathbb{E}[N(t + \delta t) - N(t_0)] \to \infty$ as $t_0 \to \infty$, for any choice of δt. This nonstationarity is easily seen by noting that, when $\alpha > 1$, each parent event produces infinitely many offspring in expectation. See [3] for further discussion.

3.2 Challenges in Direct ML Estimation

There is a convenient closed form of the log likelihood for a HP. While in principle this should enable standard 1st or 2nd-order optimization schemes for parameter estimation, in practice such methods pose severe challenges. The main problem is the low curvature near the local optimum, as shown in [13]. This low curvature leads to vanishing gradients in 1st-order methods, and severe numerical instability associated with inverting near-degenerate Hessians for second-order methods. For completeness, we introduce the likelihood function here and visualize it, before motivating an EM-based approach that circumvents these difficulties in the next subsection.

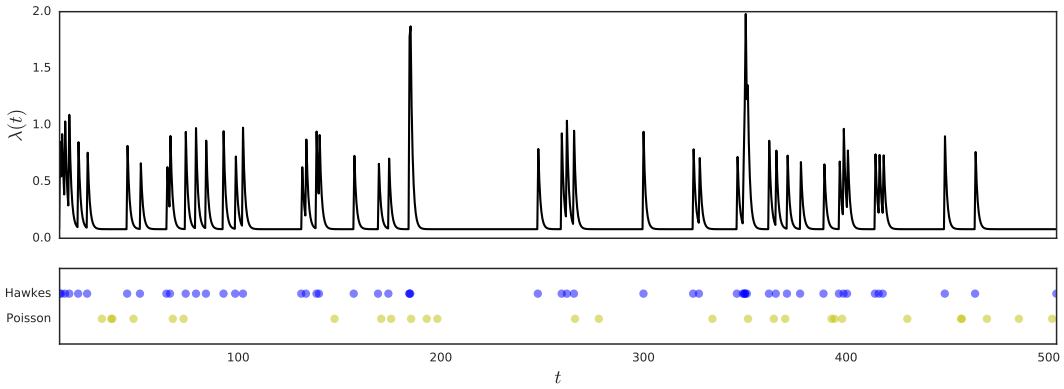


Figure 3: An example of a Hawkes process (blue dots) and its corresponding intensity function. A Poisson process (yellow dots) with rate equal to the Hawkes' base rate μ is plotted below for comparison. We can see the Hawkes process leads to temporal clustering of events not evident in the simple point process.

The likelihood of a given point process being generated by a HP with parameters μ, θ is [3, 9]:

$$\ell = \exp\left(-\int_0^T \lambda(t|\{t_j\}_{j=1}^N)dt\right)\prod_{i=1}^N \lambda\Big(t_i|\{t_j\}_{j=1}^{i-1}\Big) \tag{5}$$

which taking the log, gives

$$\log \ell = \sum_{i=1}^N \log\left(\mu + \sum_{j=1}^{i-1} g(t_i - t_j;\theta)\right) - \sum_{i=1}^N \int_0^{T-t_i} g(t;\theta)dt - \mu T$$

For an exponential triggering function of the form Eq. (4), this simplifies to the following due to Ozaki [9],

$$\log\ell\big(\{t_i\}|\theta\big) = -\mu t_N + \sum_{i=1}^N \alpha(e^{-\omega(t_N - t_i)} - 1) + \sum_{i=1}^N \log(\mu + \alpha\omega A(i))$$

with $A(i) = \sum_{t_j<t_i} e^{-\omega(t_j - t_i)}$ for $i \geq 2$, and $A(1) = 0$. However, even this special form suffers from the low curvature problem, as illustrated in Figure 4.

3.3 Regularized MAP EM for Hawkes Processes

In this section, we derive a regularized MAP EM algorithm for Hawkes process models. Our discussion is a modification of the ML approach given in [17].

Let $\tau = \{t_i\}$ denote the sequence of observed events. Let $Q = [Q_{ij}]$ be the (hidden) *branching matrix*, where $Q_{ij} = 1$ if event i is a descendant of event j. Physically, we may regard Q as encoding the unobserved, cascading causal structure of communication events. Let $p(\Theta; V)$ be a prior on the parameters Θ with hyperparameters V. We perform MAP estimation by using the EM algorithm to maximize the complete data posterior

$$p(\Theta|\tau, Q) \propto p(\tau, Q|\Theta)p(\Theta; V)\ . \tag{6}$$

Let $\mathcal{L}(\tau, Q; \Theta, V) = \log p(\tau, Q; \Theta) + \log p(\Theta; V)$ be the complete data log likelihood under the parameters Θ and hyperparameters V. The first term of $\mathcal{L}$ may be written in the form

$$\log p(\tau, Q; \Theta) = \mathcal{L}_1(\mu, \tau) + \mathcal{L}_2(\theta, \tau) + \mathcal{L}_3(\theta, \tau), \tag{7}$$

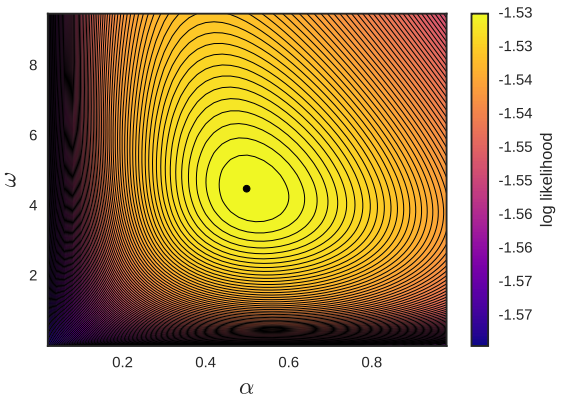


Figure 4: Contours of the log likelihood as a function of α and ω for an example cascade in the data, with fixed μ equal to its MAP estimate. The black dot in the center is the solution found via EM. While an interior optimum is clearly visible, the difference in log-likelihoods at differing parameter values is relatively small, leading to slow convergence in gradient-based schemes and numerical challenges in second-order methods.

where $\Theta = (\mu, \theta)$ and

$$\mathcal{L}_1(\mu, \tau) = -\mu T + b(\log \mu + \log T) - \log b! \tag{8}$$

$$\mathcal{L}_2(\theta, \tau) = -nG(\theta) + \sum_i d_i G(\theta) - \log d_i! \tag{9}$$

$$\mathcal{L}_3(\theta, \tau) = \sum_{ij} Q_{ij} [\log g(t_i - t_j; \theta) - \log G(\theta)] \tag{10}$$

where $b = \sum_i Q_{ii}$, $d_i = \sum_j Q_{ji}$, and $G(\theta) = \int_0^\infty g(t; \theta)\ dt$. Following some algebraic simplification, we may write

$$\begin{aligned} \log p(\tau, Q; \Theta) = &- \mu T + b \log \mu + b \log T - \log(b!) + \sum_i \left[-G(\theta) + d_i \log G(\theta) - \log(d_i!)\right] \\ &+ \sum_{ij} Q_{ij} \log g(t_i - t_j; \theta) - \log G(\theta) \end{aligned} \tag{11}$$

In the E-step of the EM algorithm, we compute a current distribution over Q. Since Q is a matrix of indicator variables, each is Bernoulli and the distribution over Q is therefore expressed by the *expected branching matrix* $P = [p_{ij}]$ based on the data τ and our current parameter estimate Θ^k. The expected branching matrix at iteration $k+1$ may therefore be computed as $P^{k+1} = \mathbb{E}[Q|\tau, \Theta^k]$. In the M-step, we update our parameter estimate to maximize the expectation of the complete data posterior log-likelihood:

$$\Theta^{k+1} = \operatorname*{argmax}_\Theta \mathbb{E}[\mathcal{L}(\tau, Q; \Theta, V)|Q = P^{k+1}] \tag{12}$$

$$= \operatorname*{argmax}_\Theta \left(\mathbb{E}[\log p(\tau, Q; \Theta)|Q = P^{k+1}] + \mathbb{E}[\log p(\Theta; V)]\right) \ . \tag{13}$$

3.3.1 Exponential Triggering and Gamma Regularization

The general MAP EM method described above may be used for arbitrary trigger functions $g(t; \theta)$ and arbitrary priors $p(\Theta; V)$. The exponential triggering function $g(t; \theta) = \alpha\omega e^{-\omega t}$ given in Eq. (4), in particular, is computationally tractable. Furthermore, as demonstrated by [17], the Hawkes process model is highly robust to the functional form of g. We therefore use the exponential triggering function, and will always assume g has the form of Eq. (4) unless explicitly stated otherwise. Recall that this function has an attractive physical interpretation: α is the *branching ratio* controlling how many subsequent events a typical event triggers, and ω is the *intensity decay rate*.

Our parameter vector is therefore $\Theta = (\mu, \alpha, \omega)$. We wish to regularize the branching ratio α and decay rate ω, but not the background intensity μ. A convenient prior on α and ω has the form

$$p(\alpha, \omega; V) = p(\alpha; V_\alpha)\ p(\omega; V_\omega) = \text{Gamma}(\alpha; s, t)\ \text{Gamma}(\omega; u, v),$$

where Gamma$(\cdot)$ is the standard gamma density with the parameterization

$$\text{Gamma}(x; a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}. \tag{14}$$

The use of the gamma prior allows us to specify both scale and location uncertainty in parameter estimates. Furthermore, as we show below, it leads to an especially simple regularization modification to the standard EM update.

3.3.2 EM Updates

We now derive the explicit forms of the EM update steps, incorporating these priors on α and ω. This extends the method outlined in [13, 4, 17].

Recall the ith event may be interpreted as either a background event or a descendant of one of the previous events. The probability that the ith event is a background event is proportional to μ^k, while the probability that it is a descendant of event j for $j < i$ is proportional to the trigger function $g(t_i - t_j; \alpha^k, \omega^k)$. The E-step update is therefore given by

$$P_{ij}^{k+1} = \begin{cases} \frac{1}{Z^k(i)} \mu^k & \text{for } i = j \\ \frac{1}{Z^k(i)} g(t_i - t_j; \alpha^k, \omega^k) & \text{for } j < i \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where the normalization is $Z^k(i) = \mu^k + \sum_{j<i} g(t_i - t_j; \alpha^k, \omega^k)$.

We now compute the M-step objective function

$$U(\Theta | P^{k+1}) \triangleq \mathbb{E}[\mathcal{L}(\tau, Q; \Theta, V) | Q = P^{k+1}],$$

which, substituting from Eqs. (11) and (14) above, is given explicitly by

$$\begin{aligned} U(\Theta | P^{k+1}) = -\mu T + \log(\mu) \sum_{P_{ii}^{k+1}} -nG(\alpha, \omega) + \sum_{i,j} P_{ij}^{k+1} \log g(t_i - t_j; \alpha, \omega) \\ + (s-1)\log\alpha - t\alpha + (u-1)\log\omega - v\omega + C(\tau, P^{k+1})\ , \end{aligned} \tag{16}$$

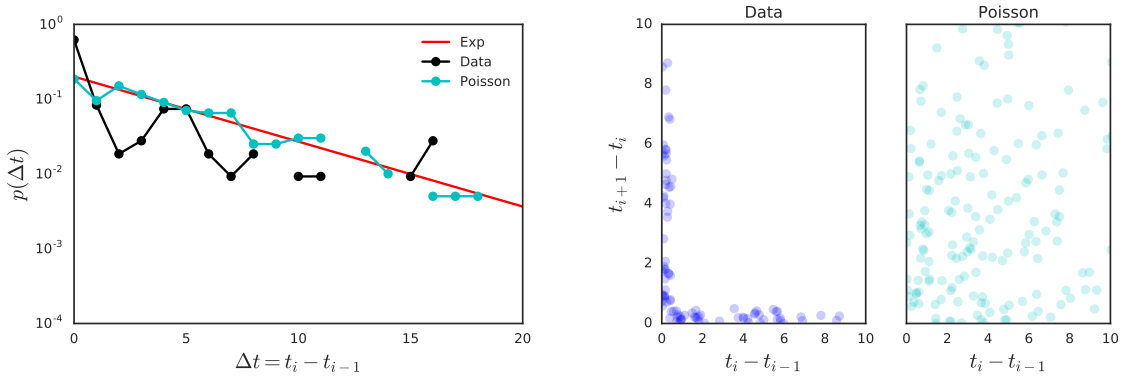
where $C(\tau, P^{k+1})$ denotes terms that do not depend on μ, α, or ω and are therefore irrelevant for parameter optimization. In the M-step, we maximize $U(\Theta | P^{k+1})$ with respect to the parameters $\Theta = (\mu, \alpha, \omega)$ The stationarity conditions $\nabla_\Theta U(\Theta | P^{k+1}) = 0$ have a unique solution, giving the M-step updates as:

$$\mu^{k+1} = \frac{1}{T} \sum_i P_{ii}^{k+1} \tag{17}$$

$$\alpha^{k+1} = \frac{1}{n+t} \left[\sum_{j<i} P_{ij}^{k+1} + s - 1 \right] \tag{18}$$

$$\omega^{k+1} = \frac{\sum_{j<i} P_{ij}^{k+1} + u - 1}{\sum_{j<i} P_{ij}^{k+1} (t_i - t_j) + v}\ . \tag{19}$$

These updates have useful interpretations that illuminate the role of the hyperparameters s, t, u, v. The first update sets μ^{k+1} equal to the expected number of background events per unit time. The second update sets α^{k+1} equal to the expected proportion of events that are descendants of a previous one, with the addition of t pseudo-observations of which $s - 1$ are descendant events. The final update sets ω^{k+1} equal to the expected number of descendant events divided by the expected total time between descent events, and therefore has the expected units of a frequency. The hyperparameter u plays the same role as s, while v may be interpreted as the total time between descendant events in the pseudo-observations. When $u = 1$ and $v = 0$ (no regularization), we can view ω^{k+1} as the reciprocal of the expected time between descendant events.



(a) (b)

Figure 5: Tests for Poisson process. In **(a)**, we compare the distribution of interarrival times in an actual sequence from the data against a Poisson sequence generated with rate equal to the average interarrival time in the data (log-lin scale). A true exponential distribution is shown as a baseline. In **(b)** we show a "lag scatter plot" of subsequent interarrival times in the Data (left) vs. a generated Poisson process (right). It is clear that while there is no correlation in the memoryless Poisson scatter, the data exhibits a clear pattern: long pauses always precede a burst of activity.

3.4 Estimated Covariance of Parameter Estimates

To estimate the covariance of the parameter estimates produced by the EM algorithm, we use a Laplace approximation of the posterior log-likelihood $\mathcal{L}$ at the optimal estimates $(\hat{\mu}, \hat{\alpha}, \hat{\omega})$. We therefore view $p(\Theta|\tau, Q)$ as locally Gaussian with mean $\hat{\Theta} = (\hat{\mu}, \hat{\alpha}, \hat{\omega})$ and covariance

$$\text{cov } \hat{\Theta} \cong \left[-\mathcal{H}_\Theta \mathcal{L}(\hat{\Theta}) \right]^{-\frac{1}{2}}, \tag{20}$$

where $\mathcal{H}_\Theta$ is the Hessian operator. Because the Hessian $\mathcal{H}_\Theta \mathcal{L}(\hat{\Theta})$ has no convenient closed form, we numerically evaluated it at the optimal parameter estimates using Python's `numdifftools` module.

4 Analysis

We will now apply the methods presented in Section 3. First, we will show justification of a non-homogeneous point process to capture the temporal clustering inherent in these conversations. Then, we will motivate our use of EM (over ML) and give a feel for its performance on some synthetic sequences. We will finally do analysis of parameter estimation both with and without regularization, such as the performance of this model as a predictor.

4.1 Test for fishiness

There are many ways of testing whether a series of points form a Poisson process. We will show two here, which albeit qualitative, give a convincing negative answer that the sequences in our data are Poissonian.

A first test is to check the distribution of the interarrival times, $\Delta t = t_i - t_{i-1}$. In a Poisson process, these are distributed $\Delta t \sim \text{Exp}(\lambda)$ for some rate λ. In Fig. 5(a) we compare the distribution of interarrival times (day scale) in an actual sequence from the data, against a generated Poisson sequence generated with the same base rate. The exponential distribution curve is shown for reference. We can see the Poisson sequence adhering to the exponential curve, while the actual data is more "bursty" — i.e. many short interarrival times, and many very long ones.

A second test is to the check the correlation in subsequent Δt, that is, the correlation between $t_i - t_{i-1}$ and $t_{i+1} - t_i$. If there is no correlation, we have reason to believe the generating process is truly "memoryless" since the Δt's appear to be independent. Fig. 5(b) shows the stark contrast between the real data and a sample Poisson process generated with the same base rate.

Taken together, these tests reassure us that there is temporal clustering occurring in the data which merits a more nuanced model.

4.2 Motivation for EM vs. ML

Now committed to modeling with the temporally clustering Hawkes process, we next justify our use of EM over a straightforward ML estimation approach. Figure 4 shows the contours of the log likelihood as a function of α and ω for an example cascade in the data (we fix μ to its ML estimate).

Although there is a clear interior optimum, we note the difference in log-likelihood at differing parameter values is quite small, so a straightforward ML approach often leads to slow convergence in gradient-based schemes. Note also that this plot shows contours with μ already fixed at the optimum, which in general will not be the case at the initial point.

On the other hand, we find in practice that the EM approach converges very quickly, and is also enticing due to the physical motivation of the branching process interpretation. This seems to give the problem additional structure which assists in the parameter estimation.

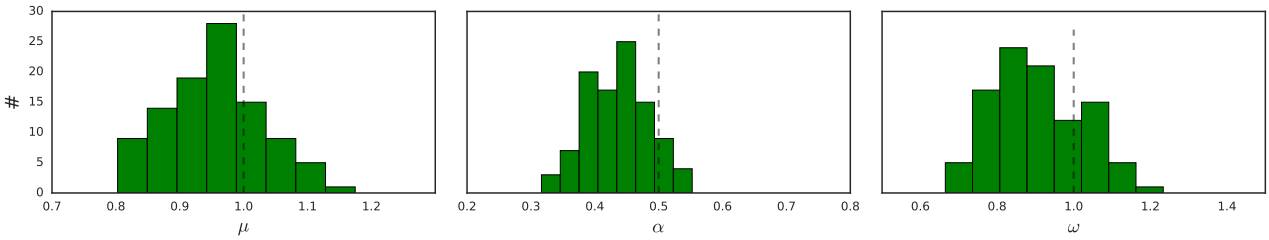


Figure 6: Histogram of estimated parameters using EM on 100 generated Hawkes processes. Ground truth values represented with a dotted black line.

4.3 EM estimation on generated sequences

Before proceeding to analysis of the actual data, let us first examine the performance of EM on some generated sequences. (We refer the reader to [3] or [17] for an overview of generating Hawkes processes. We use Lewis' "thinning method.") In this way, we can compare the estimated parameters against what we know to be "ground truth." (This replicates experiments in [13, 17].)

We generate 100 sequences over a time interval of $T = 1000$, with ground truth parameters $\mu = 1$, $\alpha = 0.5$, and $\omega = 1$. We then run EM estimation on the resulting sequences, shown in Figure 6. We find generally consistent results, but a slight leftward skew in all estimates. This variance and skew decrease as we increase the sequence size (e.g. by increasing T).

Since our data has similar number of arrivals to this generated experiment, we have reason to believe the regularization procedure (with validation selected hyperparameters) will be beneficial.

4.4 Parameter Estimation

We now investigate the results of parameter estimation using the Gamma-prior regularized MAP EM scheme outlined in Section 3.

To review, we will fit the parameters $\Theta = (\mu, \alpha, \omega)$ using the training data consisting of all sequence data before the 2-month break, and select hyperparameters $V = (s, t, v, u)$ using the validation data consisting of all sequence data after the 2-month break. We will also compare results on parameter estimates and log-likelihood both with and without this regularization step.

First, consider a comparison of the estimated parameters using EM both with and without regularization, given in Figure 7. (Each dot represents the estimate for a single sequence $\tau^{(i)}$, and the size of the dot represents the increase in log-likelihood over the minimum LL in sample).

We note the least effect on ω, which controls the decay of the triggering function, and has correlated estimates regardless of regularization. By contrast, α shows a marked separation when we introduce regularization. In particular, while under a basic MAP EM scheme we find α scattered roughly in the range 0.4 to 0.8, when

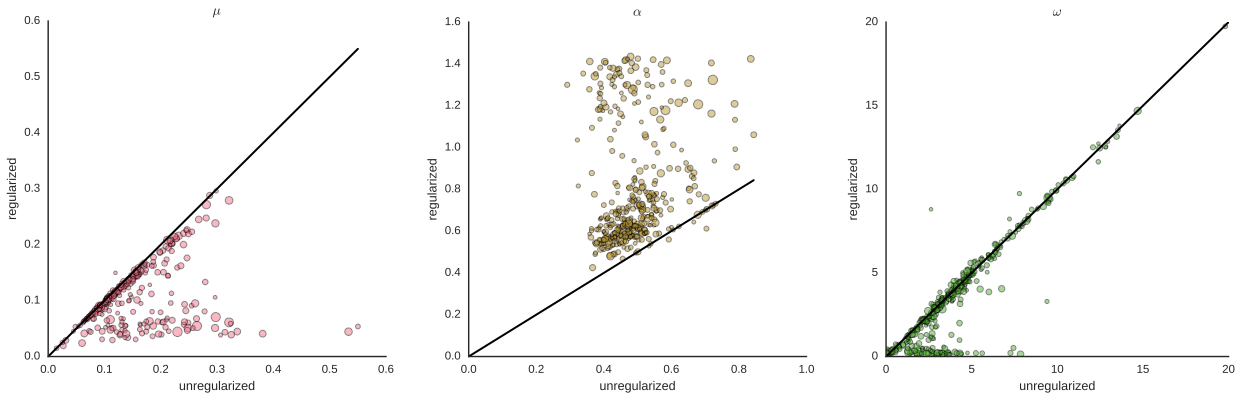


Figure 7: Comparison of parameter estimates for μ, α, ω with and without regularization. Size of dot indicates increase in validation log-likelihood over minimum LL in sample.

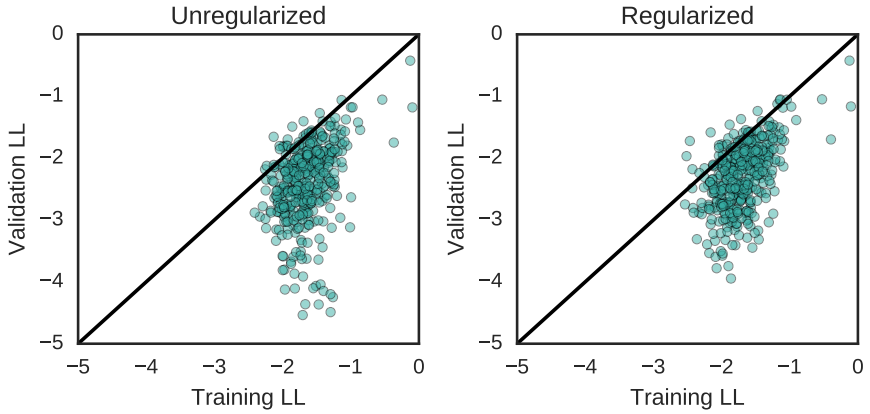


Figure 8: Scatterplots of the training log likelihood (horizontal axis) and validation log likelihood (vertical axis) for unregularized parameter estimates and optimal regularized estimates found via grid-search. Introducing validation leads to higher validation likelihoods and stronger correlation between training and validation scores.

we introduce regularization a new group of sequences emerges with $\alpha > 1.0$. This is interesting because this violates our model assumptions, that $\alpha > 1$ implies non-stationarity and a sequence that will "blow up." We investigate this further in the next subsection. Finally, we note that the regularization also appears to regress many sequences' estimate for μ back toward low values. We suspect this may also be linked to the change in α.

4.4.1 Estimate comparison and non-stationary sequences

Figure 9 shows a comparison between all three pairs of parameter estimates, which reveals some of the dynamics at play. Note that in these plots, the dot size indicates the *size* of the sequence, $|\tau^{(i)}|$.

We first note the general trend of positive correlation in the last ω vs. μ plot, which indicates that as the base rate leads to more and more expected arrivals, the effect of each arrival tends to decrease. We also note that this is not limited to longer sequences, where we might expect the effect to be necessary to prevent the sequence blowing up, but even in short sequences.

We now consider the first plot, of α against μ, that the cluster of sequences with non-stationary α also has a much lower μ than the rest of the data. This indicates that the sequences simply have a large number of events, and instead of capturing this with a high base intensity μ, the optimization is using a non-stationary α. This is interesting, since it indicates that a highly temporally clustered process (that is, higher α) is still a better predictor in this case than a simple process with high intensity.

The second plot also shows this non-stationary group behaving with different dynamics as relates to ω —

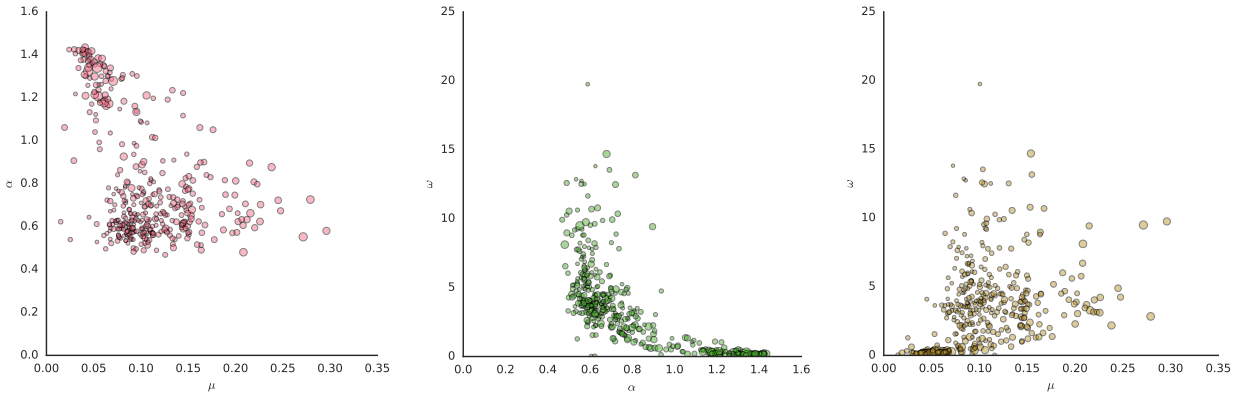


Figure 9: Correlation scatters of parameter estimates for μ, α, ω under regularization. Size of dot here indicates size of the sequence.

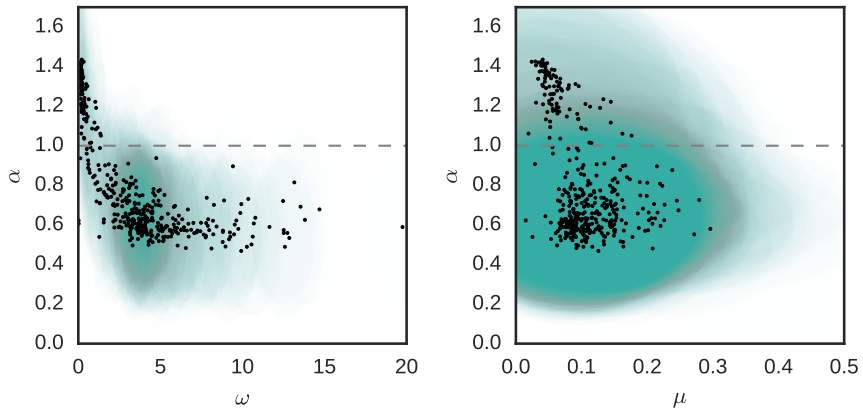


Figure 10: Bivariate scatters of parameter estimates for α, ω, and μ. Around each parameter estimate is drawn a standard-deviational ellipse estimated using the Laplace approximation given by (20). The dashed line indicates the threshold $\alpha = 1$ above which the branching process is nonstationary.

the non-stationary group has very low values of trigger function decay, which is surprising as we might expect the ω parameter to "compensate" for the high branching ratio by being even *higher*.

Figure 11 shows three example sequences from the data, with respectively low, median, and high estimated values of α. The non-stationarity of the third sequence ($\alpha = 1.44$) is reflected in the fact that the intensity is almost never at its baseline value. We also see the slow decay exhibited in this process observed in the previous plot.

4.4.2 Estimator covariance

Figure 10 shows the parameter estimates superimposed on standard deviational ellipses computed using (20). We observe relatively high covariance in both stationary and nonstationary estimates, reflecting both the volatility of the point process model and the relatively small amount of data contained in each cascade. Future research may explore the availability of larger samples or the appropriateness of parameter-sharing among multiple cascades according to flexibly-defined grouping criteria.

4.4.3 Effect of regularization on validation performance

Figure 8 illustrates the effect of the Gamma prior regularization on performance in the validation set. In particular, we note that using optimal hyperparameters in regularization (obtained through grid-search) corrects

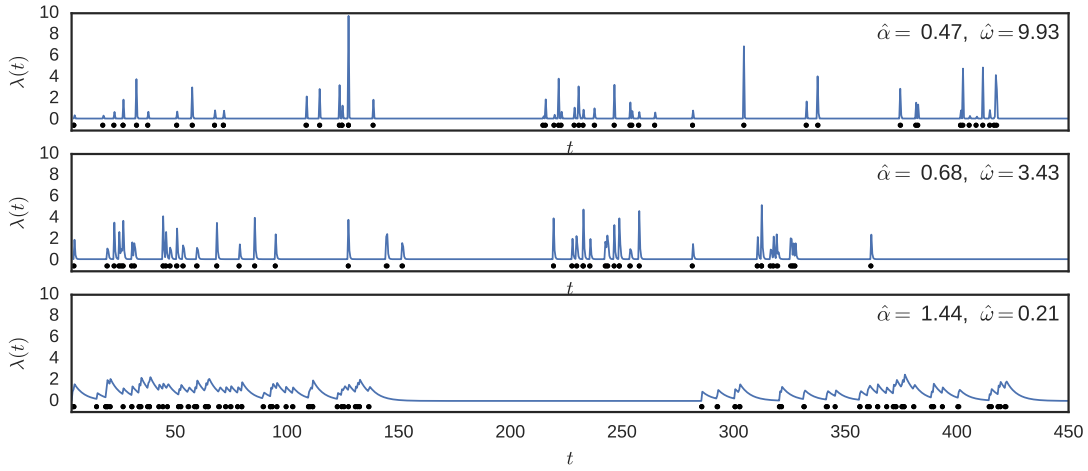


Figure 11: Process events (black dots) and estimated intensities using MAP parameters for sequences with low (top), median (middle) and high (bottom) estimated branching ratios $\hat{\alpha}$. The nonstationarity of the third sequence is reflected in the fact that the intensity is almost never at its baseline value.

overfitting on a large group of sequences and creates stronger correlation between training and validation scores.

5 Conclusion and future work

We have shown that certain persistent group conversations between individuals in a communication network are by nature temporally clustered and better modeled by a non-homogeneous point process than, for example, a Poisson process. We have introduced a regularized MAP EM scheme for estimating parameters under such a model, using a Gamma prior and validation-selected hyperparameters. We illustrated that this scheme works will and produces interpretable results, despite relatively small and somewhat noisy datasets. It also shows that many real sequences in the data generate what appear to be non-stationary processes, violating a necessary model assumption.

This leads us to future work. The non-stationarity found may be due to the construction of the cascades, which requires that all events fall within a pre-defined time interval. This creates perhaps unnecessarily dense temporal clustering effects — there are likely "follow-on" events outside the time interval that are not captured, and would contribute to possibly relaxed values of α and ω, under the interpretation presented in this project.

As a result, we are interested in generalizing our approach to recovering all mutually exciting relationships in the network, similar to work in [6] and [16] as mentioned in the Introduction. It may be possible to use the persistent structure gained from our current preliminary analysis as a "prior" to guide the network-wide search in the cited papers, which suffer from a dimensionality and complexity limitation.

Division of labor. S. Morse wrote the code which extracted the sequences from the unprocessed data. P. Chodrow derived the regularized version of MAP EM. Both authors contributed equally to data analysis, plot production, and writeup.

6 Appendix: Persistent Cascades

6.1 Definition.

Consider a temporal graph $G = (V, E)$ which represents the communications between users over some large time period $T = [t_{\text{begin}}, t_{\text{end}}]$, such as one month. Let each node $v \in V$ represent a user who participates in some number of communication events during period T, and let each edge $e \in E$ represent a communication event which we encode as a 4-tuple $e_i = (s_i, d_i, t_i, \delta_i)$ consisting of the initiator (s_i), the receiver (d_i), the time of the event (t_i), and its duration (δ_i).

We define a *time-respecting* path as any sequence of edges $(e_1, e_2, ..., e_k)$ such that for any consecutive pair e_i, e_j in the sequence, we have that $d_i = s_j$ and $t_i + \delta_i \leq t_j$. We define a Δt-*connected* path as a time-respecting path such that $t_k - t_1 \leq \Delta t$. From these definitions, one can construct Δt-connected subgraphs that contain some time-respecting subset of *all* the events within Δt.

However, in pursuit of understanding information spread patterns, we make an assumption that the information *originates from a single user*, and every user receives the information at the *earliest possible time*. This implies there is a single in-edge to each user, and creates a rooted, directed tree structure. Intuitively, this shifts focus from the structure of the call patterns to the structure of the information spread, since we will only capture the first occurrence of "information" being passed.

Formally, this assumption leads to the construction of a rooted, directed, Δt-connected tree which we term a *cascade*. This term, and its construction, follows closely that in [10].

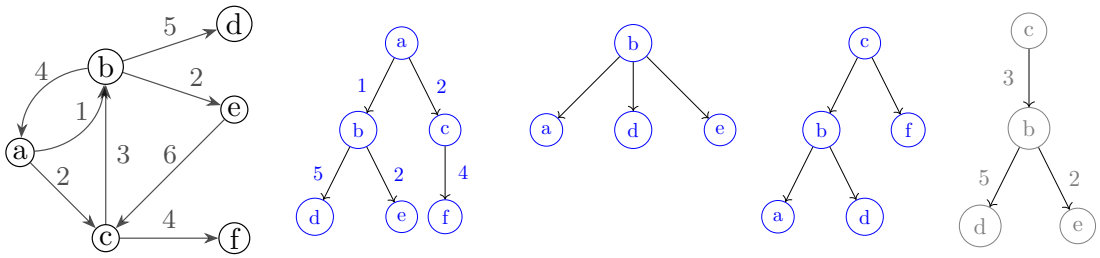
Denote a cascade with root r as C_r, denote the set of all cascades for root r with maximum time interval Δt and total time period T as $\mathcal{C}_r(T, \Delta t)$, and use superscripts as necessary to distinguish multiple cascades with the same root. For example, we might have the set of all cascades for some root a:

$$\mathcal{C}_a(T = 1 \text{ mo},\ \Delta t = 24 \text{ hrs}) = \{C_a^1, C_a^2, C_a^3\} \tag{21}$$

Note we require that the intervals not overlap: i.e. no calls from C_a^1 can also be in C_a^2, etc.

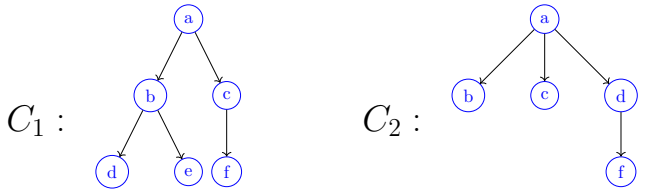
An example of cascade construction from a network with all temporal information is shown in Figure 12

Figure 12: Simplified illustration of cascade extraction from a temporal graph. For clarity, we examine a network with only 6 nodes. **(a)** Full temporal information ($\Delta t = 6$ units, times depicted on edges). **(b)** Three valid cascades given this temporal snapshot. Note that there is no time ordering of children within a cascade. **(c)** Invalid cascade because: (c-b-e) is not a time-connected path, and missing the edge (c-f).



6.2 Similarity measures.

Tree edit distance Edit distance is the process of counting the minimum number of insertions, deletions, or mutations required to transform one string into another. One can extend this concept to trees. Denote the tree edit distance between two trees (or cascades) C_1 and C_2 as $\text{TED}(C_1, C_2)$, which maps two cascades to a nonnegative integer. As an example, consider the following two trees :



To change C_1 into C_2, we can delete d and e, mutate c into d, and add c again, giving $\text{TED}(C_1, C_2) = 4$ (note this is the same to change C_2 into C_1).

A canonical algorithm for computing this distance is due to Zhang and Shasha ([15]), which we implement with the `zss` package available at `https://github.com/timtadh/zhang-shasha`. We can now define a similarity measure using this distance as follows.

Definition 6.1. *Tree Edit Distance similarity.* Define the normalized tree edit similarity as

$$s_{\text{TED}}(C_1, C_2) \stackrel{\text{def}}{=} 1 - \frac{2 \cdot \text{TED}(C_1, C_2)}{|C_1| + |C_2| + \text{TED}(C_1, C_2)}. \tag{22}$$

and note s_{TED} lies on $[0, 1]$.

This definition is due to [5], who also prove that the corresponding distance metric $1 - s_{\text{NTED}}$ meets the triangle inequality. Note we make every edit operation unit cost.

Using the example trees above, we now compute $s_{\text{TED}} = 1 - \frac{2 \cdot 4}{6+5+4} = \frac{7}{15} \approx 0.47$.

Reach set Consider the un-ordered set of all nodes in a tree. For a cascade, this corresponds to all users who the root reached during the time period Δt, and potentially received some information. We term this the *reach set* of a cascade (similar to concepts in [LI PAN]).

A simple first approximation of the similarity of two cascades is by comparing their reach sets. Let $R(C_i)$ denote the reach set of a cascade C_i. Now, given two cascades C_1 and C_2, define the similarity measure s_{RS} as the Jaccard index of the two reach sets, that is

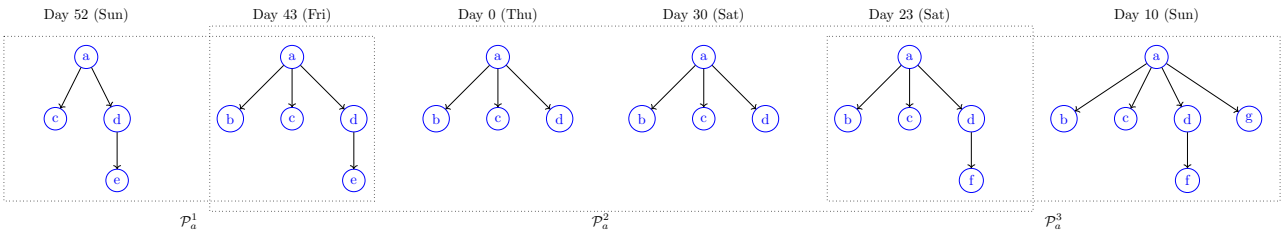
Definition 6.2. *Reach Set similarity.* Given two cascades C_1 and C_2, and their reach sets $R(C_1)$ and $R(C_2)$, define

$$s_{\text{RS}}(C_1, C_2) \stackrel{\text{def}}{=} \frac{|R(C_1) \cap R(C_2)|}{|R(C_1) \cup R(C_2)|}. \tag{23}$$

and note s_{RS} lies on $[0, 1]$.

Continuing with the previous example, we have $s_{\text{RS}}(C_1, C_2) = \frac{5}{6} \approx 0.83$.

Figure 13: Actual set of cascades for a root a over a 60-day period. Six persistent cascades are shown, each from temporal subgraphs with $\Delta t = 24$ hours. Dotted rectangles depict the persistence class groupings. We see a clear set of "core friends" (nodes b, c, d), and slight variations incorporating other groups. We also see the overlap that occurs when a cascade appears to fit in multiple classes. Labeled above each cascade is the day of the week.



6.3 Persistence

We now would like to group cascades together which all share some minimum pairwise similarity, and so are in a relaxed (but well-defined) sense the "same cascade." This group now represents various incarnations of some fundamental communication structure. We call these groups *persistence classes*, and the elements of each group *persistent cascades*, and they are the main object of our analysis.

Definition 6.3. *Persistence class.* Define the i-th persistence class of root r, similarity threshold ℓ in time period T over intervals Δt, as the set

$$\mathcal{P}_r^i(\ell, T, \Delta t) = \left\{ C_r^1, C_r^2 \in \mathcal{C}_r(T, \Delta t) \ : \ s_*(C_r^1, C_r^2) \geq \ell \right\} \tag{24}$$

and the collection of all persistence classes for a particular root as $\mathcal{P}_r(\ell, T, \Delta t)$.

Definition 6.4. *Persistent cascade.* Define a persistent cascade as any cascade C_r^i such that $C_r^i \in \mathcal{P}_r(\cdot)$, for some r.

Note we may also choose to ignore any persistence classes below a certain size. The minimum size is 2 by construction, but we may decide based on the parameters T and Δt that a minimum size of 3 or more is appropriate.

To find these classes, our definition and Eq. (24) leads us directly to an agglomerative clustering approach with complete-linkage — that is, define the similarity between two clusters U and V as $s(U, V) = \min s_*(U_i, V_j), \forall i \in U, \forall j \in V$ where U_i, V_j represent cascades within U and V. Then the clusters at iteration k, such that every pairwise similarity within the cluster is $\geq s_k$, represent persistence classes with $\ell = s_k$.

However, this assumes that each cascade falls uniquely into one class, which we can imagine is not always true: a spreading pattern among work friends may overlap with the pattern among social friends, and there may be cascades that are not clearly in one class or the other.

So we instead adopt a graph-theoretic interpretation of the complete-linkage approach: represent each data point (cascade) as a vertex in a graph $H(s_k)$ such that each any two vertices with similarity $\geq s_k$ are connected. Then the clusters at iteration k correspond to the maximal completely connected subgraphs in H, also known as the maximal cliques.

Now, applying this technique, consider the collection of persistence classes $\mathcal{P}_a$ depicted in Figure 13, taken from City A. Here, we see a core pattern consisting of root a calling b, c, and d, captured in $\mathcal{P}_a^2$. Then, we see two variations on this core structure: $\mathcal{P}_a^1$ which incorporates e, and $\mathcal{P}_a^3$ which incorporates f and g. Since they are mostly weekend calls, we might easily imagine this being a core group of social friends, with variations possibly for family or work acquaintances.

References

[1] Barabasi, A.-L. The origin of bursts and heavy tails in human dynamics. *Nature 435* (2005).

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

[3] Laub, P. J., Taimre, T., and Pollett, P. K. Hawkes Processes. *arXiv.org math.PR* (2015).

[4] Lewis, E., and Mohler, G. A Nonparametric EM algorithm for Multiscale Hawkes Processes. *Journal of nonparametric statistics*, 1 (2011), 1–20.

[5] Li, Y., and Zhang, C. A metric normalization of tree edit distance. *Front. Computing Sci. 5*, 1 (2011), 119–125.

[6] Linderman, S. W., and Adams, R. P. Discovering Latent Network Structure in Point Process Data. *arXiv preprint 32* (2014), 1413–1421.

[7] Miritello, G., Moro, E., and Lara, R. Dynamical strength of social ties in information spreading. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* (2011).

[8] Morse, S., Gonzalez, M. C., and Markuzon, N. Persistent Cascades : Measuring Fundamental Communication Structure in Social Networks. In *Proc. IEEE Big Data Conf* (2016).

[9] Ozaki, T. Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annual Institute of Statistical Mathematics 31*, B (1979), 145–155.

[10] Peruani, F., and Tabourier, L. Directedness of Information Flow in Mobile Phone Communication Networks. *PloS one 6*, 12 (2011).

[11] Steglich, C., Snijders, T. A. B., and Pearson, M. Dynamic Networks And Behavior: Seperating Selection From Influence. *Sociological Methodology 8* (2010), 329–393.

[12] Vazquez, A., Rácz, B., Lukács, A., and Barabási, A. L. Impact of non-poissonian activity patterns on spreading processes. *Physical Review Letters* (2007).

[13] Veen, A., and Schoenberg, F. P. Estimation of Space–Time Branching Process Models in Seismology Using an EM-Type Algorithm. *Journal of the American Statistical Association 103*, 482 (2008), 614–624.

[14] Watts, D. J., Dodds, P. S., and Newman, M. E. J. Identity and Search in Social Networks. *Science 296*, May (2002), 1302–1305.

[15] Zhang, K., and Shasha, D. SIMPLE FAST ALGORITHMS FOR THE EDITING DISTANCE BETWEEN TREES AND RELATED PROBLEMS*. *SIAM J. Computing 18*, 6 (1989), 1245–1262.

[16] Zhou, K., Zha, H., and Song, L. Learning Social Infectivity in Sparse Low-rank Networks Using Multi-dimensional Hawkes Processes. *... of the Sixteenth International Conference on ...* (2013), 641–649.

[17] Zipkin, J. R., Schoenberg, F., Coronges, K., and Bertozzi, A. Point-process models of social network interactions : parameter estimation and missing data recovery. *Euro Journal of Applied Mathematics 1* (2014).