Nested CRP with Hawkes-Gaussian Processes

Xi Tan[1], **Vinayak Rao**[2], **and Jennifer Neville**[1,2]
[1]Department of Computer Science and [2]Department of Statistics
Purdue University

Abstract

There has been growing interest in learning social structure underlying interaction data, especially when such data consist of both temporal and textual information. In this paper, we propose a novel nonparametric Bayesian model that incorporates senders and receivers of messages into a hierarchical structure that governs the content and reciprocity of communications. We bring the nested Chinese restaurant process from nonparametric Bayesian statistics to Hawkes process models of point pattern data. By modeling senders and receivers in such a hierarchical framework, we are better able to make inferences about the authorship and audience of communications, as well as individual behavior such as favorite collaborators and top-pick words. Empirical results show that our proposed model has improved predictions about event times and clusters. In addition, the latent structure revealed by our model provides a useful qualitative understanding of the data, facilitating interesting exploratory analyses.

1 INTRODUCTION

Communication on social networks tends to exhibit clustering both in time and content, and quantifying this phenomenon has been a subject of long interest in social sciences. Early work in the machine learning community [5, 13, 28] used declared relationships between entities to infer hidden clusters, however such data are usually hard to obtain, and the declared relationships themselves are far from reliable. Instead, interaction data have been used to learn latent structure in an unsupervised manner. Of particular relevance is the work of [7], that combined the Infinite Relational Model (IRM) [5, 28] and Hawkes processes [10, 11] to learn social structure from interaction data. The benefit of using Hawkes processes are two-fold: first they capture the self- and mutually-exciting temporal dynamics of communication activities, and second, their probabilistic nature enables the introduction of rich structure into the modeling.

While providing mechanistic models of interacting point pattern data, the original Hawkes models do not account for message content: for social media data, this is clearly an important factor determining how one event affects future activity. Recent work in this direction includes [8, 9, 16]. Modeling message content accurately at the individual level involves identifying and exploiting latent hierarchical structure present among users, and we exploit ideas from nonparametric Bayes to improve the relatively impoverished structure present in earlier work.

Formally, we consider an underlying network where nodes are interacting entities, with communication events forming links in the network. The observed data $\mathcal{D}$ consists of a sequence of n messages $\mathcal{D} = (M_1, \cdots, M_n)$, sorted by their time stamps. Each message M_i is a quadruplet $M_i = \{t_i, S_i, R_i, \mathcal{T}_i\}$, where t_i is a time-stamp, S_i the set of senders, R_i the set of receivers, and $\mathcal{T}_i$ the content of the message. Note that we allow multiple senders (e.g., in modeling citation networks) and multiple receivers (e.g., in modeling email data). We are interested in the following tasks:

- At the node level, we would like to learn a hierarchical clustering $\mathcal{C}$ for all the entities in the network, such that entities in the same cluster share some common features of communication including rates, content, collaborators, audiences, etc.

- At the link level, given previous activity $\mathcal{D}$, we would like to predict the message quadruplet $M_{n+1}|\mathcal{D} = (t_{n+1}, S_{n+1}, R_{n+1}, \mathcal{T}_{n+1})|\mathcal{D}$, both at the cluster level and at the individual level. Realistic modeling of $\mathcal{T}_{n+1}$ requires sophisticated language models, which is not our focus. Instead, we

Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

are interested in demonstrating how incorporating hierarchical structure at the node level significantly improves predictions of message time and content. Accordingly, we limit ourselves to predicting keywords in user messages, rather than detailed message content.

Main Contributions:

- We introduce senders (S_i) and receivers (R_i) into a novel and unified framework combining the advantages of hierarchical nonparametric Bayesian models and temporal point processes. This enables us to leverage temporal (t_i) and textual ($\mathcal{T}_i$) information present in the communications, allowing improved predictions about event times and clusters.
- Our method exploits senders' and receivers' properties to characterize message content, enabling inference about authorship and audience of communications, as well as their personal behavior such as favorite collaborators and top-pick words.

2 PRELIMINARIES

We start with a brief description of Hawkes processes (HPs), the Chinese restaurant processes (CRP), and its nested version, the nested CRP (nCRP).

Hawkes Processes: One of the most powerful and popular temporal point process models is the inhomogeneous Poisson process, parametrized by a rate function $\lambda(t)$, which is independent from its history events. In real-world social network communications however, messages directly and causally affect each other. Poisson processes cannot capture such self- or mutual-excitation, and instead, there has been much interest in using Hawkes processes (HPs) to model such data. At a high-level, a self-exciting Hawkes process [10] has a rate-function that is dependent on its own history (i.e., $\lambda(t)$ is dependent on the event history for $s \leq t$). Similarly, a pair of mutually-exciting Hawkes processes have mutually-dependent rate functions that depend on each others' histories.

Formally, let $N(\cdot)$ and $N'(\cdot)$ be counting measures representing a pair of mutually-exciting Hawkes processes. The conditional rate function $\lambda(t)$ of $N(\cdot)$, given the event time history $\mathcal{H}_{N'} = \{t'_1, \cdots, t'_n\}$ of N', has the form:

$$\lambda(t) = \gamma + \int_{-\infty}^{t} g(t-s)\, dN'(s) \tag{1}$$

Here γ is the base rate of $N(\cdot)$, and the *excitation function* $g(\cdot)$ is a non-negative function such that $\int_0^\infty g(s)\, ds < 1$, ensuring the stationarity of $N(\cdot)$. A standard choice for g is the exponential function, which implies that every event from N' produces a jump in the intensity $\lambda(t)$, which then decays exponentially to the base rate. If the counting measure $N'(\cdot)$ is $N(\cdot)$ itself, then the process is self-exciting. The likelihood function of a Hawkes process, given conditional rate function $\lambda(t)$ and event time history $\mathcal{H}_{(0,T]} = (t_1, \cdots, t_n)$, is

$$\mathcal{L}(\lambda(t)|\mathcal{H}) = \exp\{-\Lambda(0,T)\} \prod_{i=1}^{n} \lambda(t_i) \tag{2}$$

where $\Lambda(0,T) = \int_0^T \lambda(t)\, dt$ is the cumulative conditional rate function.

The Chinese Restaurant Process (CRP) and its nested version (nCRP): The CRP is an infinitely exchangeable probability distribution over partitions that can be described using the following metaphor involving customers entering a restaurant: The first customer sits at table 1; the following customers pick a new table with probability proportional to some constant, and pick an existing table with probability proportional to the number of people already assigned to that table:

$$p(\pi_i|\pi_{-i}) = \begin{cases} \frac{\alpha}{N-1+\alpha} & \text{if } \pi_i \text{ a new table} \\ \\ \frac{|B_j|}{N-1+\alpha} & \text{if } \pi_i \text{ an existing table } j \end{cases} \tag{3}$$

where π_{-i} is the assignment vector π without the i^{th} entry, and $|B_j|$ is the number of customers seated at table j. The joint probability is $p(\pi|\alpha) = \alpha^{|B|} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \prod_{j=1}^{|B|} \Gamma(|B_j|)$, where $|B|$ is the total number of tables, and $(|B_j|-1)!$ is the factorial of $|B_j|-1$, the number of individuals in the j^{th} table minus one.

The nested Chinese Restaurant Process (nCRP) is similar to a CRP, but with a hierarchical tree structure (see Figure 1a). For an nCRP with L levels, rather than being assigned to a single table, a user is assigned to a sequence of L tables. After a customer comes into the first restaurant and picks a table, the customer is directed to a level-2 restaurant, again picking tables according to the paths of previous users. This process repeats $L-1$ times until the customer finds a seat at a level-L restaurant. The consequence now is that a customer selects not just one table, but a sequence of tables; in our application, this will allow a message to belong not just to a user or group, but a nested set of groups. For more details on the nCRP, see [4, 6].

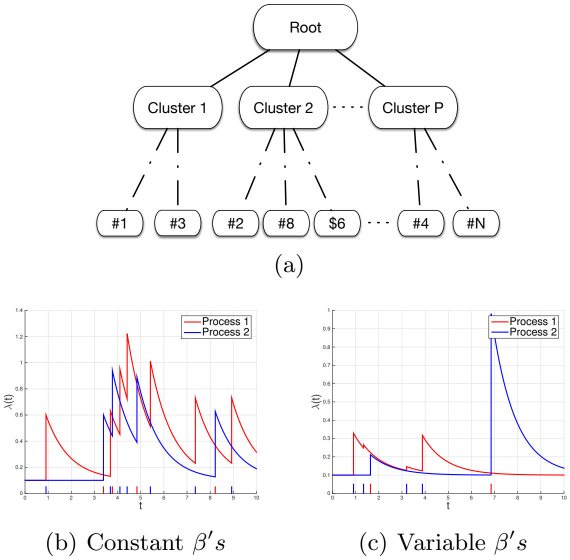
3 MODEL

Since every piece of information in our data is indexed by time, modeling t_i is of central importance. Recall that if we only have one individual, the form of a

Hawkes process with an exponential-decay excitation function g is given by:

$$\lambda(t) = \gamma + \int_{-\infty}^{t} \beta e^{-\frac{t-s}{\tau}} \, \mathrm{d}\, N(s) \tag{4}$$

The parameter β can be seen as a "jump size" of the rate function whenever a new message is received (see Figure 1), and τ indicates the *inverse* rate of decaying.



(a)

(b) Constant $\beta's$

(c) Variable $\beta's$

Figure 1: (a) A clustering tree sampled from nCRP, (b-c) Hawkes processes rate function plots with constant and variable $\beta's$.

To incorporate text information, we first allow the jump sizes to depend on the message content via a function $\beta : \Re \rightarrow \Re$. The function β takes some feature of the message $\mathcal{T}_i$ as input (e.g. the entropy of the message), and determines the size of the Hawkes excitation. We model β with a Gaussian Process (GP) [22]:

$$\lambda(t) = \gamma + \int_{-\infty}^{t} \beta(f(\mathcal{T}_s)) e^{-\frac{t-s}{\tau}} dN(s) \tag{5}$$

$$\beta(f(\mathcal{T}_i)) \sim \exp(\mathcal{GP}(0, \kappa)) \tag{6}$$

where $\mathcal{T}_i$ is the text communicated at t_i, $f(\cdot)$ some transformation that converts text content into numerical measurement, κ the squared exponential kernel of the GP, and the exponential transformation is used to make sure that $\beta(\cdot)$ is non-negative. While there are many ways to implement the transformation $f(\mathcal{T}_i)$, we propose the following: 1) calculate TF-IDF scores for each word in the message $\mathcal{T}_i$, so that the sentence is represented by a vector; 2) from their vector representations, calculate distances between pairs of sentences in the message; 3) use the TextRank [18] algorithm to pick the top sentences and summarize a top-word distribution; 4) compute the KL-divergence between this top-word distribution and the personalized word distribution of the individual. Effectively, this allows us to quantify how 'relevant' each message is to the receiver.

3.1 Modeling Senders and Receivers (S_i, R_i)

Now suppose we have multiple individuals, and a *flat* (one level) clustering $\mathcal{C}$. We define the rate function between two individuals u and v as

$$\lambda_{uv}(t) = \frac{1}{n_p n_q} \gamma_{pq} + \int_{-\infty}^{t} \beta_{uv} e^{-\frac{t-s}{\tau_{uv}}} dN_{vu}(s) \tag{7}$$

where u and v belong to clusters p and q respectively, and n_p, n_q the number of individuals in clusters p and q. The subscript ordering of N_{vu} (instead of N_{uv}) indicates these Hawkes processes are mutually exciting. Unlike work in [7], which models rates at the cluster level, we model rate functions at the individual level. The benefits of this are three-fold: first, individuals in the same cluster share common behavior through cluster level parameters γ_{pq}; second, unlike cluster-level models (which uniformly pick individuals from a cluster), we explicitly model activity at the individual level; and finally, we need not separately define cluster level rate functions. Instead, the latter can be computed as sums of individual rate functions:

$$\lambda_{pq}(t) = \sum_{p=\pi(u), q=\pi(v)} \lambda_{uv}(t) \tag{8}$$

where $\pi(u)$ is the cluster assignment of individual u. To select senders and receivers from clusters, define the *unconditional* cumulative rate of a sender u, and the *conditional* cumulative rate of a receiver v of a message from a set of senders S as

$$\bar{\lambda}_{u\cdot}(t) = \sum_{v} \lambda_{uv}(t), \quad \bar{\lambda}_{\cdot v|S}(t) = \sum_{u \in S} \lambda_{uv}(t). \tag{9}$$

Then the probabilities of u and v respectively being selected as one of the receivers and senders are proportional to their cumulative rate ratios:

$$Z_{u \in S} \sim Ber\left(\frac{\bar{\lambda}_{u\cdot}(t)}{\sum_u \bar{\lambda}_{u\cdot}(t)}\right) \tag{10}$$

$$Z_{v \in R|S} \sim Ber\left(\frac{\bar{\lambda}_{\cdot v|S}(t)}{\sum_v \bar{\lambda}_{\cdot v|S}(t)}\right) \tag{11}$$

where $Z_{u \in S}$ and $Z_{v \in R|S}$ are indicator variables that u and v are selected. The receivers are conditionally picked *after* the selection of senders.

3.2 The Overall Model

Recall that at the node level, we would like to learn, not a flat, but a hierarchical tree-like clustering for all the individuals in a network. We model this as a sample from a nested Chinese restaurant process. Conditioned on this tree, it is straightforward to compute all the rates in a *bottom-up* fashion, by summing up the rates, level by level, all the way from the leaf nodes (individuals), using equation 8. Based on these rates, senders and receivers can be selected recursively in a *top-down* fashion, using equations 10 and 11.

The generative process of our model works as follows: 0) sample a clustering tree from the nCRP prior; 1) based on historical data $\mathcal{D} = (M_1, \cdots, M_n)$, compute the rate at the root by summing up over all relevant lower level rates (at the beginning, we only have the base rates γ_{pq}); 2) simulate a new event time t_{n+1} based on the root rate; 3) select senders S_{n+1} and receivers R_{n+1} of *each level* of the clustering tree for this new message (the real senders and receivers will be the ones at the leaf level); 4) generate the message text $\mathcal{T}_{n+1}$ from a multinomial distribution based on senders S_{n+1} and receivers R_{n+1} at the leaf level; 5) finally, update the rate functions of all the receivers. Thus we have generated $M_{n+1} = (t_{n+1}, S_{n+1}, R_{n+1}, \mathcal{T}_{n+1})|\mathcal{D}$. Repeat steps 1) through 5) with $\mathcal{D} = (M_1, \cdots, M_n, M_{n+1})$. This can be summarized as:

$$\pi|\alpha \sim nCRP(\alpha) \tag{12}$$

$$\lambda_{uv}(t) = \frac{1}{n_p n_q}\gamma_{pq} + \int_{-\infty}^{t} \beta_{uv}(f(\mathcal{T}_s))e^{-\frac{t-s}{\tau_{uv}}} dN_{vu}(s) \tag{13}$$

$$\lambda_{pq}(t) = \sum_{p=\pi(u),q=\pi(v)} \lambda_{uv}(t) \tag{14}$$

$$M_{new} = \begin{cases} t_{new} \sim HawkesProcess(\lambda_{root}(\cdot)) \\ Z_{u\in S_{new}} \sim Ber\left(\frac{\bar{\lambda}_{u\cdot}(t_{new})}{\sum_u \bar{\lambda}_{u\cdot}(t_{new})}\right) \\ Z_{v\in R_{new}|S_{new}} \sim Ber\left(\frac{\bar{\lambda}_{\cdot v|S}(t_{new})}{\sum_v \bar{\lambda}_{\cdot v|S}(t_{new})}\right) \\ \mathcal{T}_{new} \sim Multinomial(\theta_{S_{new},R_{new}}) \end{cases} \tag{15}$$

where nCRP is the nested Chinese Restaurant Process, and $\beta_{uv}(f(\mathcal{T}_i)) \sim \exp(\mathcal{GP}(0, \kappa_{uv}))$.

The texts are generated from multinomial distributions whose parameters depend on the senders and receivers: We add and normalize the individual word distributions of the senders and receivers and use the aggregated one for the multinomial distribution.

3.3 Inference

Inference algorithms for Hawkes processes fall mainly into three categories [12]: 1) methods related to Maximum Likelihood Estimation (MLE) [21], which are usually quite restrictive and incompatible with rich latent structure; 2) variational approximations [26], which often suffer from poor convergence issues and are best applicable when the inference problem exhibits a convenient simplifying approximation; and 3) Monte Carlo sampling methods.

For our model, the inference problem is nonparametric and non-convex, and there is no conjugacy between the priors and the likelihood functions. We therefore adopt and extend the inference framework from [7] and [24], which performs posterior inference using MCMC sampling. The state space of the model is defined over $\{\pi_u, \gamma_{uv}, \tau_{uv}, \beta_{uv}, \theta_u\}$, and the conditional distributions used in the MCMC algorithm can be obtained based on section 3.2. The sketch of the algorithm can be described as follows: 1) Initialize the state variables by sampling from their priors. 2) Until convergence, iteratively and sequentially sample each state variable conditioned on the current state of all other variables – sample π_u using the standard Gibbs sampling algorithm [6]; sample $\{\theta_u, \gamma_{uv}, \tau_{uv}\}$ using slice sampling [20]; sample β_{uv} using elliptical slice sampling [19].

For a dataset of N individuals, M messages, and K top words, the number of model parameters is $\mathcal{O}(N^2)$, and the computational cost at each iteration is $\mathcal{O}(MN^2K^3)$. One of the bottlenecks of the algorithm comes from the inference of the GP related parameters β_{uv}, which costs $\mathcal{O}(K^3)$, where K is the number of top words. To ameliorate this situation, we restrict K to be a reasonably small number in our experiments, e.g., $K = 20$. We also want to point out that, at each iteration, not all of the $O(N^2)$ parameters are updated or used to update other parameters. For example, after an update of π_u, only the affected individuals and clusters should be considered – which is usually a small subset of the population in practice.

4 RELATED WORK

The closest existing work to our model are [4, 8, 24], though none of these explore hierarchical clusterings of senders and receivers with Hawkes processes. The model of [4] combines ideas from the hierarchical Dirichlet process (HDP) [25] and the nested Chinese Restaurant Process (nCRP) [3] to allow each object to be represented as a mixture of paths over a tree, and to decouple the task of modeling hierarchical structure from that of modeling observations. The work of [8] connects Dirichlet processes and Hawkes processes to allow the number of clusters to grow while at the same time learning the changing latent dynamics governing the continuous arrival patterns. The combination of these two pieces of work inspired our work, which has a hierarchical structure embedded with temporal point processes.

Recently, [14, 15, 16, 17, 23, 26, 27] proposed different models to address similar problems. However, while we define each observed message M_i as a quadruplet $M_i = \{t_i, S_i, R_i, \mathcal{T}_i\}$, these previous work, in our opinion, all missed some important aspects of the information. The loss of these may result in ineffectiveness of modeling personal level details. For example, [17] modeled $M = \{t, S, R\}$, [16, 23, 26] modeled $M = \{t, S, T\}$, and [15] modeled $M = \{t, S\}$ and the cluster $\mathcal{C}$. Our work explicitly treats senders $\{S_i\}$ and receivers $\{R_i\}$ as important components of the model, which greatly extends the existing methods in the literature and enables inference about authorship and audience of communications, as well as their personal behavior such as their favorite collaborators and top-pick words.

Moreover, we focus on different modeling perspectives, specifically, (1) modeling mutually-exciting transactions between users (e.g., email communications) rather than individual self-exciting actions of users (e.g., purchases/clicks), and (2) modeling personalized textual content between pairs of users (with a continuous metric), rather than modeling individual topics/tasks (with a discrete metric). While topics/tasks can be viewed as discrete labels of the "content" of activities (and it is meaningful to use this concept in cases such as web activities), in the context of communications/transactions, the content being communicated is highly personalized, a continuous metric affords more flexibility to make better use of personalized content.

5 EXPERIMENTS

We compare our model with four existing models (discussed in Sections 1 and 4): nCRP+HP, CRP+HGP, IRM+HP, and HP. Recall that IRM stands for the infinite relational model, HP for the Hawkes process and HGP for the Hawkes process with a Gaussian process controling jumps. We first present experimental results based on synthetic data, which focus on quantitative analysis of model performance as well as qualitative discussions of model effectiveness. We then explore some of the findings from real data using our model. The observed data $\mathcal{D}$ used in this section has the same format, consisting of a sequence of messages $\mathcal{D} = (M_1, \cdots, M_n)$, sorted by their time stamps. Each message M_i is a quadruplet $M_i = \{t_i, S_i, R_i, \mathcal{T}_i\}$, where t_i is the time-stamp, S_i the set of senders, R_i the set of receivers, and $\mathcal{T}_i$ the text content of the message. $\mathcal{D}$ is divided into three segments: the first 80% the training set, the next 10% the validation set, and the last 10% the test set. To compute the average log probability, we run each experiment ten times with different prior settings and report the credible interval based on their means and standard deviations.

5.1 Synthetic Data

Following the generative process described in Section 3.2, we simulate 1000 message communications among 7 individuals (shown in Figure 2). The clustering tree has two levels, $\{\#1, \#2, \#3\}$ are in cluster 1 (red), $\{\#4, \#5\}$ in cluster 2 (green), and $\{\#6, \#7\}$ in cluster 3 (blue). The initial rate γ at the root is set to 1, and this is distributed among its offspring proportional to their cluster sizes following Equation 1. The inverse decay rates τ_{uv} are set to 0.1 for all pairs of u, v. The "jump size" function is taken to be an exponential $\beta(x) = \exp(x)$. The vocabulary of the synthetic corpus we used consisted of the top 10,000 words from the Neural Information Processing Systems (NIPS) dataset (consisting of 5811 papers published during the years 1987 to 2015). We generate 1000 messages, each containing 20 words. The personalized distributions over the 10,000 words of the seven users are randomly generated through a Dirichlet distribution, the concentration parameters of which are drawn from a Dirichlet prior with uniform concentration parameters.

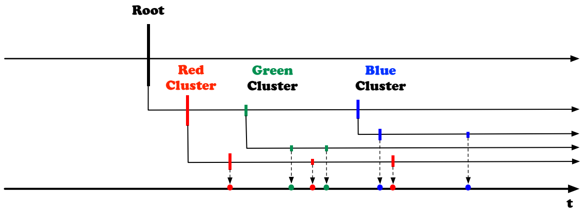


Figure 2: nCRP + HGP plot. The clustering tree has two levels (root is at level 0): the first level consists of three clusters (red, green, and blue), and at the second level each of the cluster has several individuals (red cluster has 3 individuals, green has 2, and blue has 3). Individuals *receive* messages (represented by color dots) at different times, which bump the rate functions of individuals (represented by color bars) by a certain amount (decided by the GPs). The heights of the bars at the cluster level and at the root illustrate the aggregate effect from lower level rates.

Predictive log-likelihood. We compare our method with the alternatives, showing results in Table 1. We see that our model achieved the best performance in terms of predictive log-likelihood. This is not surprising, given that the data is generated from the model.

	Predictive Log-likelihood
nCRP + HGP	312.89 (± 12.37)
nCRP + HP	221.97 (± 10.16)
CRP + HGP	207.63 (± 13.28)
IRM + HP	197.23 (± 16.12)
HP	101.01 (± 16.12)

Table 1: nCRP+HGP against other models. Log-likelihoods with standard deviations (10 runs).

The three main components of our model are: 1) GP to model varying "jump sizes"; 2) nCRP for hierarchical clustering; and 3) senders and receivers to model personalized textual information. We investigate the effectiveness of these model characteristics.

Usefulness and identifiability of the GPs. In Table 1, we already see that nCRP+HGP had higher log-likelihoods compared to nCRP+HP, suggesting that including the GPs helps our models overall predictive performance. Here, we take a closer look at the actual fit of each GP compared to the ground truth (the exponential). Shown below are the GP plots of the first three of the seven individuals (along with the truth), showing the ability of the GPs to recover the underlying "jump size" function $\beta(x)$.

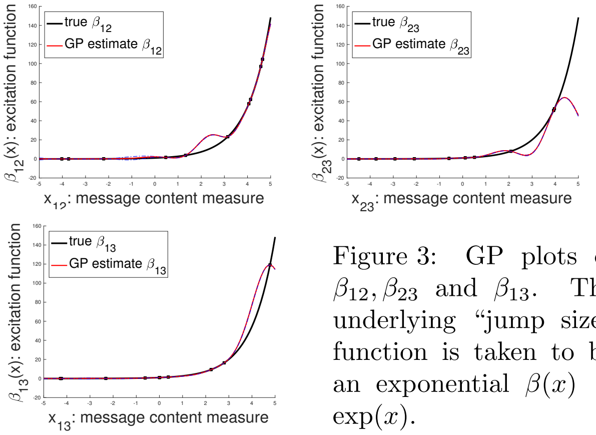


Figure 3: GP plots of β_{12}, β_{23} and β_{13}. The underlying "jump size" function is taken to be an exponential $\beta(x) = \exp(x)$.

Effect of nCRP for modeling hierarchical clustering structure. We compare our model with two manually designed trees: (i) the true underlying tree; (ii) an incorrect tree that puts all 7 individuals in one single cluster. Our model which samples trees from nCRP prior recovers the tree structure. From Table 2 we see that it obtained very similar predictive log-likelihood as that based on a correct manual tree, compared to the much worse performance using an incorrect manual tree. The correct manual tree achieves smaller standard deviation over 10 experiment runs, which is what we expected since the fixed tree reduces randomness of the model. It is also clear that ignoring the tree results in poor predictive log-likelihood.

	Predictive Log-likelihood
(nCRP) sampled tree	312.89 (± 12.37)
(correct) manual tree	321.92 (± 7.86)
(incorrect) manual tree	126.27 (± 21.63)
no tree	179.61 (± 9.17)

Table 2: Sampled trees against manual trees. Log-likelihoods with standard deviations (10 runs).

Benefits of including senders and receivers. One of the advantages of introducing senders and receivers is the ability to generalize the thinning procedure in Hawkes processes. In previous work (e.g., [7]), uniform thinning is a popular choice, i.e. a new message is assigned to an individual with equal probability. Our model on the other hand can assign a message to its senders and receivers based on 1) its event history, via the HPs; 2) text information, via the GPs; and 3) collaborator and audience, via the nCRPs.

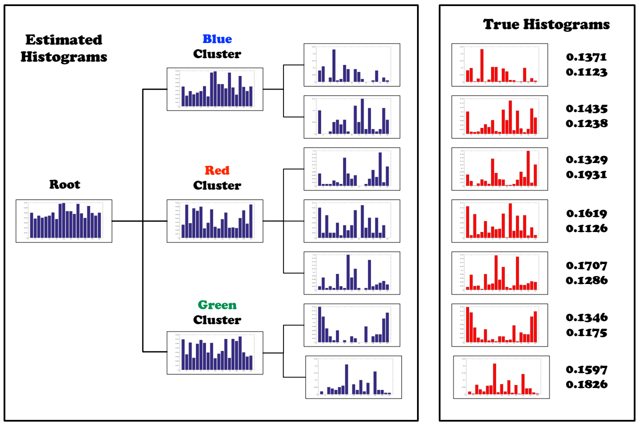


Figure 4: Posterior keyword distributions of synthetic dataset. The first numbers are the *estimated* word distributions at each node on the nCRP tree; and the second numbers are the *true* word distributions, together with their L_1 distances (against top 20 words and 20,000 full vocabulary).

To demonstrate these benefits, the final experiment on synthetic data focuses on learning the posterior keyword distributions of individuals, which may be used to suggest personalized favorite words, and in turn decide the authorship and audiences of the new messages.

The leaf nodes in Figure 4 shows the posterior keyword distributions of the seven individuals. The cluster level keyword distribution is aggregated from its members' distributions (top words of the union of top words), and the root keyword distribution is aggregated from the cluster ones. Thus, the top words in each histogram may not be the same. We also notice that at the root, the words are almost uniformly distributed, which suggests that the most important words across all individuals are almost of the same importance. We may use these top words to identify clusters.

5.2 Real Data

We apply our method to three different real datasets:

NIPS Dataset [1]. This contains the counts of 11,463 words appearing in the 5,811 papers published in the conference Neural Information Processing Systems (NIPS) during the years 1987 to 2015. Authors and citations are obtained through the paper IDs. We treated authors as message "senders", and cited authors as "receivers".

Facebook Dataset. This data contains Facebook message communications among 20,603 individuals. We pick the top 10 individuals based on their number of friends, and add in their 1st and 2nd connection friends (376 in total).

Santa Barbara Corpus Dataset [2]. The Santa Barbara Corpus [2] dataset (SB) contains text recordings for various conversations. The data we use (#33) is a lively family discussion recorded at a vacation home in Falmouth, Massachusetts. There are eight participants, all relatives or close friends. Discussion centers around a disagreement that Jennifer (#2) is having with her mother Lisbeth (#5).

Predictive log-likelihood. We evaluate our model performance in terms of predictive log-likelihood, and present our findings about keywords and clusters. For all of these three datasets, the predictive log-likelihoods of our model constantly outperform existing alternative methods.

	NIPS Dataset
nCRP+HGP	9708.23, 1297.83, **1127.21**
nCRP+HP	9026.78, 1028.36, **997.82**
CRP+HGP	8934.67, 1186.22, **1128.76**
IRM+HP	4896.17, 567.18, **682.70**
HP	3490.78, 518.70, **683.18**
	Facebook Dataset
nCRP+HGP	1208.37, 199.12, **218.93**
nCRP+HP	992.70, 181.11, **178.86**
CRP+HGP	1118.61, 175.81, **182.49**
IRM+HP	928.14, 128.76, **129.83**
HP	312.78, 59.08, **61.93**
	Santa Barbara Dataset
nCRP+HGP	491.37, 118.12, **109.82**
nCRP+HP	391.87, 96.24, **99.68**
CRP+HGP	438.71, 101.83, **97.20**
IRM+HP	412.98, 81.87, **52.73**
HP	303.82, 59.83, **70.23**

Table 3: Model comparison on the real datasets. The numbers reported in each cell are the log-likelihoods for training, validation, and **test** set, respectively.

Next, we show the effectiveness and consistency of our model, i.e., what our model can do with different types of datasets and whether or not it gives us consistent performance under different scenarios.

Exploratory analysis. *1) Identifying clusters and learning interesting community features.* Figure 5 shows the posterior word distribution at the root node for the **Facebook dataset**. The size of each word is proportional to its "importance", based on the TF-IDF scores. We see that: firstly, the sizes are quite uniform, agreeing with our findings from synthetic data analysis; and secondly, the words with highest "importance" are "happy" and "birthday", confirming the 'viral" nature of mutually-exciting Hawkes processes. We also summarize the sizes of the first two clusters, as well as top 3 words of each cluster. Cluster 1 has 128 individuals, with top 3 keywords {*workout, class, homework*}; Cluster 2 has 95 individuals, with top 3 keywords {*time, work, break*}. Based on the keywords, we suggest that cluster 1 is more about study and school life, cluster 2 is more about work, and related activity.

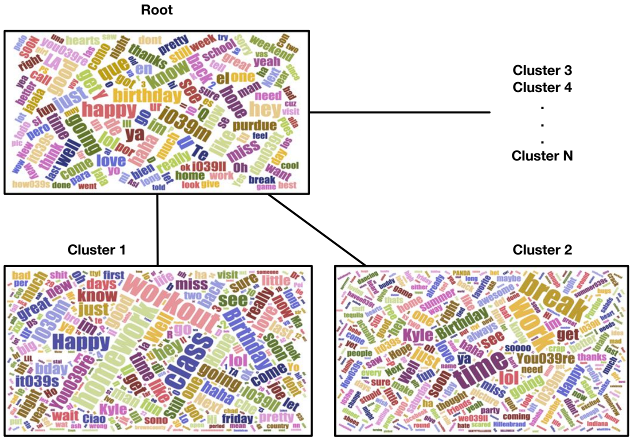


Figure 5: Facebook data WordCloud.

2) Predicting preferences of senders/receivers within each cluster. Shown below are the predicted collaborators and keywords of three selected top authors (in terms of number of papers and citations) from the **NIPS dataset**. This clearly aligns with what we know about the authors' research interests. These predicted preferences of individuals play an important role in deciding the authorship and patterns of future communications.

- *Y. Bengio (+ G. Hinton, Y. LeCun):* deep learning, neural network, data, machine learning, features, gradient.
- *Z. Ghahramani (+ M. Jordan, D. Blei):* neural network, kernel, variational, probabilistic, Gaussian processes, regression.
- *Y. LeCun (+G. Hinton, Y. Bengio):* generative, embedding space, auto-encoder, supervised.

3) Interpret individual behavior via quantifiable evidence. Figure 6 shows the rate function plots of two clusters from the **Santa Barbara dataset**: Jennifer and her mother Lisbeth, and the rest of the people. We see that there is a trend that whenever topic 1 (between Jennifer and her mother Lisbeth) is active, topic 2 tends to become silent. This phenomenon is clearly observed during (normalized) time frame 70 to 90. The actual transcript of this conversation shows that this was one of the occasions when Jennifer and Lisbeth were arguing with each other. It is even clearer when we look closer at the rate functions at the individual level. Figure 6 shows Jennifer and Lisbeth's individual rate functions, which are almost complement to each other.

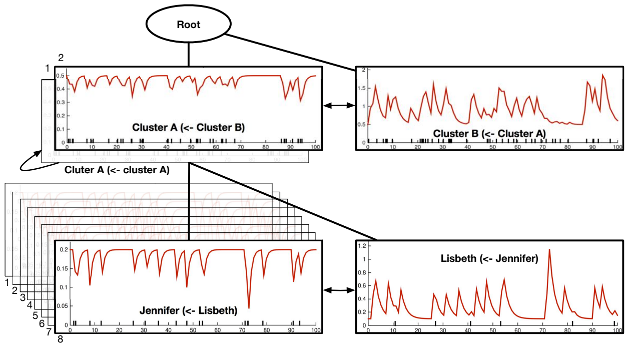


Figure 6: Rate function plots of the SB data at the cluster level: {A: Jennifer and Lisbeth} and {B: Others}; and individual level. At the individual level, there are eight rate functions associated with each person (only shown Jennifer in the plot), including the one with him/herself. Cluster rates are aggregations of individual rates, as defined in equation 8.

Learning parameters with an incorrect tree. To evaluate the importance of jointly learning the tree structure from the data, we shuffle the tree and relearn the parameters and compare the log-likelihoods as follows: 1) Learn a tree $\mathcal{T}$ from the model; 2) shuffle nodes to obtain a new tree $\mathcal{T}'$; and then 3) use $\mathcal{T}'$ and re-learn the parameters. Repeat the process ten times and report mean and standard deviation.

	NIPS Dataset
model	9708.23, 1297.83, **1127.21**
without a tree	8934.67, 1186.22, **1128.76**
bottom level (leaves)	3790.41±130.19, 489.23±79.81 **414.98±27.37**
bottom 2 levels	1279.83±189.76, 316.78±88.61, **316.78±28.72**
bottom 3 levels	997.81±212.86, 283.68±107.75, **278.91±30.67**
	Facebook Dataset
model	1208.37, 199.12, **218.93**
without a tree	1118.61, 175.81, **182.49**
bottom level (leaves)	216.16±29.78, 37.65±7.63, **67.54±9.82**
bottom 2 levels	186.72±31.78, 21.98±9.27, **51.28±10.67**
bottom 3 levels	121.67±36.15, 21.45±10.62, **45.27±12.19**
	Santa Barbara Dataset
model	491.37, 118.12, **109.82**
without a tree	438.71, 101.83, **97.20**
bottom level (leaves)	278.23±12.96, 79.81±9.71, **87.15±7.12**
bottom 2 levels	212.67±9.18, 71.93±12.38, **72.85±10.37**
bottom 3 levels	217.56±18.92, 68.73±17.92, **67.17±16.84**

Table 4: Log-likelihood comparison after shuffling the tree from the model, under different depth. The numbers reported in each cell are the log-likelihoods for training, validation, and **test** datasets, with their standard deviations, respectively.

In Table 4, our model outperform the ones without a tree and shuffled-trees, and in particular, the more we destroy the structure of the tree, the worse the model performance. This confirms that our model's superior performance is not because of the additional parameters from the tree: it is the tree structure itself that is important.

Model comparisons. For each real dataset, we divide the dataset into 10 equal-length pieces $D_1, D_2, \cdots, D_{10}$, and then perform an increasing-size training strategy: use D_1 to train the model and test on D_{10}; use D_1 and D_2 for training and test on D_{10}; and so on, until finally, train model using $D_1, \cdots, D_9$ and test on D_{10}. The results in Figure 7 suggest that our model consistently outperforms other models in the comparison, especially in its ability to learn better at early stages with relatively small amounts of data. For larger amounts of data, the model without the tree structure performs comparably, which explains some of the results in Table 4.

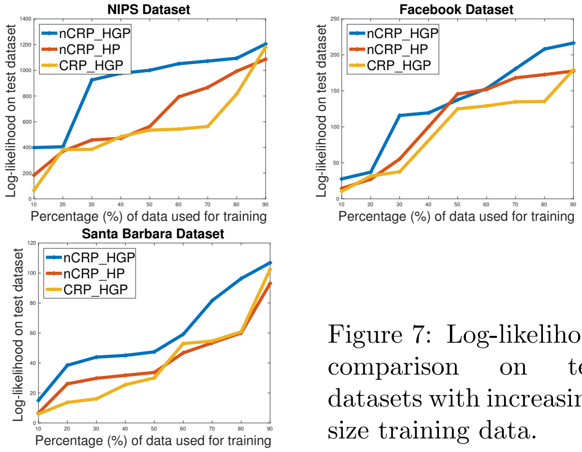


Figure 7: Log-likelihood comparison on test datasets with increasing-size training data.

6 CONCLUSION

In this paper, we have established a novel and unified framework combining the advantages of Bayesian nonparametrics and temporal point processes to model not only the temporal (t_i) and textual ($\mathcal{T}_i$) information of the messages communicated in a network, but also the senders (S_i) and receivers (R_i) who are involved in the communications. Empirical results suggest that our novel model formulation can improve predictions about event times, clusters, etc. In addition, our method offers inference about authorship and the audience of communications, as well as their personal behavior such as their favorite collaborators and top-pick words, which greatly extends the existing methods in the literature.

Acknowledgements

This research is supported by NSF under contracts IIS-1546488 and IIS-1618690. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of NSF or the U.S. Government.

References

[1] NIPS Dataset. http://archive.ics.uci.edu/ml/.

[2] SB Dataset. www.linguistics.ucsb.edu/research/santa-barbara-corpus.

[3] A. AHmed and E. Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *SIAM International Conference on Data Mining (SDM)*, 2010.

[4] Amr Ahmed, Liangjie Hong, and Alexander J. Smola. Nested chinese restaurant franchise processes: Applications to user tracking and document modeling. In *International Conference on Machine Learning (ICML)*, 2013.

[5] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research (JMLR)*, 2008.

[6] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[7] Charles Blundell, Jeff Beck, and Katherine A Heller. Modeling reciprocating relationships with hawkes processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[8] Nan Du, Mehrdad farajtabar, Amr Ahmed, Alexander J. Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

[9] Fangjian Guo, Charles Blundell, Hanna Wallach, and Katherine Heller. The bayesian echo chamber: Modeling social influence via linguistic accommodation. In *Artificial Intelligence and Statistics (AISTATS)*, 2015.

[10] Alan G. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1971.

[11] Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 1971.

[12] Niao He Jalal Etesami, Negar Kiyavash and Yingxiang Yang. Online learning for multivariate hawkes processes. In *Neural Information Processing Systems (NIPS)*, 2017.

[13] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2006.

[14] Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *International Conference on Machine Learning (ICML)*, 2014.

[15] Le Song Hongyuan Zha Long Tran, Mehrdad Farajtabar. Netcodec: Community detection from individual activities. In *IEEE International Conference on Data Mining (ICDM)*, 2015.

[16] Charalampos Mavroforakis, Isabel Valera, and Manuel Gomez-Rodriguez. Modeling the dynamics of learning activity on the web. In *International Conference on World Wide Web (WWW)*, 2017.

[17] Manuel Gomez Rodriguez Shuang Li Hongyuan Zha Le Song Mehrdad Farajtabar, Yichen Wang. Coevolve: A joint point process model for information diffusion and network coevolution. In *Neural Information Processing Systems (NIPS)*, 2015.

[18] R. Mihalcea and P. Tarau. Textrank bringing order into texts. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.

[19] Iain Murray, Ryan P. Adams, and David Mackay. Elliptical slice sampling. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[20] Radford M. Neal. Slice sampling. In *Annals of Statistics*, 2003.

[21] Tohru Ozaki. Maximum likelihood estimation of hawkes self-exciting point processes. In *Annals of the Institute of Statistical Mathematics*, 1979.

[22] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[23] Soheil Arabzade Seyed Abbas Hosseini, Ali Khodadadi and Hamid R. Rabiee. Hnp3: A hierarchical nonparametric point process for modeling content diffusion over social media. In *IEEE International Conference on Data Mining (ICDM)*, 2016.

[24] Xi Tan, Syed A. Z. Naqvi, Katherine A. Heller Yuan (Alan) Qi, and Vinayak Rao. Content-based modeling of reciprocal relationships using hawkes and gaussian processes. In *Uncertainty in Artificial Intelligence (UAI)*, 2016.

[25] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet

processes. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[26] James Foulds Lise Getoor Xinran He, Theodoros Rekatsinas and Yan Liu. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In *International Conference on Machine Learning (ICML)*, 2015.

[27] Hongteng Xu and Hongyuan Zha. A dirichlet mixture model of hawkes processes for event sequence clustering. In *Neural Information Processing Systems (NIPS)*, 2017.

[28] Z Xu, V Tresp, K Yu, and HP Kriegel. Infinite hidden relational models. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.