

The Hyperedge Event Model

Bomin Kim*, Aaron Schein†, Bruce A. Desmarais‡, and Hanna Wallach§

Abstract. We introduce the hyperedge event model (HEM)—a generative model for events that can be represented as directed edges with one sender and one or more receivers or one receiver and one or more senders. To define the model, we integrate a dynamic version of the exponential random graph model (ERGM) of edge structure with a survival model for event timing to jointly understand who interacts with whom, and when. The HEM offers three innovations with respect to the literature—first, it extends a growing class of dynamic network models to model hyperedges. Second, we derive a receiver-selection distribution that forces the sender to select at least one receiver. Third, we incorporate both a timing equation and an edge-formation equation into the model specification. We use the HEM to analyze emails sent among department managers in Montgomery County government in North Carolina. Our application demonstrates that the model is effective at predicting and explaining time-stamped network data involving edges with multiple receivers. We present a out-of-sample prediction experiment to illustrate how researchers can select between different specifications of the model.

MSC 2010 subject classifications: Primary 60K35, 60K35; secondary 60K35.

Keywords: sample, L^AT_EX 2_&.

1 Introduction

In recent decades, real-time digitized textual communication has developed into a ubiquitous form of social and professional interaction (Kanungo and Jain, 2008; Szóstek, 2011; Burgess et al., 2004; Pew, 2016). From the perspective of the computational social scientist, this has led to a growing need for methods of modeling interactions that manifest as edges exchanged in continuous time. In recent decades, several dynamic network models which could handle time-stamped events with temporal granularity have been developed, where their edge formations are governed by structural dynamics similar to those used in the exponential random graph model (ERGM)—the canonical model for modeling the structure of a static network. For example, stochastic actor-oriented models (SAOMs) (Snijders, 1996; Snijders et al., 2007) study the network evolutions as a consequence of *micro-steps*—i.e., actors making choices to form a new tie or withdraw an existing tie. On the other hand, event-based network models (Butts, 2008; Vu et al., 2011; Hunter et al., 2011) provide a general regression framework for longitudinal networks by modeling an edge as a realization of continuous time survival process. Those models are flexible enough to specify a generative model that accounts for nearly any pattern of edge formation such as reciprocity, clustering, popularity effects (Desmarais

*Department of Statistics, Pennsylvania State University bzk147@psu.edu

†College of Information and Computer Sciences, UMass Amherst aschein@cs.umass.edu

‡Department of Political Science, Pennsylvania State University bdesmarais@psu.edu

§ Microsoft Research NYC bdesmarais@psu.edu

and Cranmer, 2017), thus very useful to understand which traits and behaviours are predictive of interactions.

However, most studies above are limited to modeling edges with one sender and one receivers. To our knowledge, none of the existing continuous-time network models in statistical literature explicitly allow hyperedges—a connection between two or more vertices generated simultaneously¹—which is a common property of digitalized textual interactions such as emails and online messages. For instance, Perry and Wolfe (2013) treat multicast interactions—one type of directed hyperedge which involve one sender and one or more receivers—via duplication (i.e., obtain pairwise interactions from the original multicast), to construct approximate likelihood function in their inferential framework for model parameters. Similarly, Fan and Shelton (2009) duplicates emails sent from one sender to one or more receivers and randomly jitter the sent times, in order to avoid the violation of continuous time model assumption. There have been numerous studies for hyperedges under the hypergraph theory (Karypis et al., 1999) in physics literature, including a random hypergraph model (Ghoshal et al., 2009), but most of them are designed specifically for engineering applications—e.g., multiple tagging networks (Zlatić et al., 2009; Zhang and Liu, 2010), cellular networks (Klamt et al., 2009), and personalized recommendations (Zhang et al., 2010; Blattner, 2009)—with lack of extensive efforts to incorporate hyperedge modeling into general statistical framework.

Concentrating on developing a statistical network model with better handling of hyperedges, we introduce the hyperedge event model (HEM) which jointly models the two components that govern time-stamped event formation: 1) the receiver selection process that allows directed edges with one sender and one or more receivers or one receiver and one or more senders, and 2) the timing equation that models time increments (i.e., time to next event) with flexible distributional choices. In what follows, we introduce the HEM by describing how we assume the generative process of a time-stamped event data (Section 2), and deriving the conditional posteriors for Bayesian inference along with code review tests (Section 3). Then, we demonstrate the model’s applicability by analyzing one of the county government email network—the Montgomery county email data—via model selection, posterior predictive checks, and exploratory analysis (Section 4). Finally, the paper finishes in Section 5 with conclusion and discussion.

2 The Hyperedge Event Model

Data generated under the model consists of D unique edges. A single edge, indexed by $d \in [D]$, is represented by the three components: a sender $a_d \in [A]$, an indicator vector of receivers $\mathbf{r}_d = \{u_{dr}\}_{r=1}^A$, and a timestamp $t_d \in (0, \infty)$. For simplicity, we assume that edges are ordered by time such that $t_d \leq t_{d+1}$. While the model can be applied for two type of hyperedges—edges with (1) one sender and one or more receivers, and (2) one receiver and one or more receivers—here we only present the generative process for those involving one sender and one or more receivers (i.e., multicast). For the latter case

¹A hyperedge connecting just two nodes is simply a usual dyadic edge.

of hyperedges, we treat a_d to be an indicator vector of senders $\mathbf{a}_d = \{u_{dr}\}_{r=1}^A$ and r_d to be the single receiver. Detailed generative process for directed edges with one receiver and one or more senders are provided in Appendix A.

2.1 Hypothetical Edges

For every possible sender–receiver pair $(a, r)_{a \neq r}$, we define the “receiver intensity”—an approximate inverse logit of the probability that edge d is being sent from sender a to receiver r —as a linear combination of statistics relevant to the receiver selection process:

$$\lambda_{adr} = \mathbf{b}^\top \mathbf{x}_{adr}, \quad (2.1)$$

where \mathbf{b} is a P -dimensional vector of coefficients and \mathbf{x}_{adr} is a set of receiver selection features which vary depending on the hypotheses regarding canonical processes relevant to network theory such as popularity, reciprocity, and transitivity. In addition, we include intercept term to account for the average (or baseline) number of receivers. We place a Normal prior $\mathbf{b} \sim N(\boldsymbol{\mu}_b, \Sigma_b)$.

Next, we hypothesize “If a were the sender of edge d , who would be the receivers?” and derive a receiver-selection distribution. For an edge d , we first define an $A \times A$ matrix \mathbf{u}_d where the a^{th} row denotes sender a ’s receiver vector of indicators—i.e., if node r is the hypothetical receiver of sender a , then $u_{adr} = 1$; otherwise $u_{adr} = 0$. We then assume that each receiver vector \mathbf{u}_{ad} comes from the multivariate Bernoulli (MB) distribution (Dai et al., 2013)—a model to estimate the structure of graphs with binary nodes—with logit probability of $\boldsymbol{\lambda}_{ad}$. In order to avoid the model degeneracy from having an empty receiver set, we define a probability measure “ MB_G ” motivated by the non-empty Gibbs measure (Fellows and Handcock, 2017) which excludes the all-zero vector from the support of the multivariate Bernoulli distribution. As a result, this measure helps us to 1) allow multiple sender–receiver pairs to co-exist, 2) force the sender to select at least one receiver, and 3) ensure tractable normalizing constant. To be specific, we draw a binary vector $\mathbf{u}_{ad} = (u_{ad1}, \dots, u_{adA})$

$$\mathbf{u}_{ad} \sim \text{MB}_G(\boldsymbol{\lambda}_{ad}), \quad (2.2)$$

where $\boldsymbol{\lambda}_{ad} = \{\lambda_{adr}\}_{r=1}^A$. In particular, we define $\text{MB}_G(\boldsymbol{\lambda}_{ad})$ as

$$\Pr(\mathbf{u}_{ad} | \mathbf{b}, \mathbf{x}_{ad}) = \frac{1}{Z(\boldsymbol{\lambda}_{ad})} \exp \left(\log(I(\|\mathbf{u}_{ad}\|_1 > 0)) + \sum_{r \neq a} \lambda_{adr} u_{adr} \right), \quad (2.3)$$

where $Z(\boldsymbol{\lambda}_{ad}) = \prod_{r \neq a} (\exp(\lambda_{adr}) + 1) - 1$ is the normalizing constant and $\|\cdot\|_1$ is the l_1 -norm. Again, this is equivalent to assuming independent Bernoulli trial on each u_{adr} with probability of 1 being $\text{logit}(\lambda_{adr})$, excluding the case when $u_{adr} = 0$ for all $r \in [A]$. We provide detailed derivation steps for the normalizing constant in Appendix B.

2.2 Hypothetical Timestamps

Similarly, we consider “If a were the sender of edge d , when would it be sent?” and define the “timing rate” for sender a

$$\mu_{ad} = g^{-1}(\boldsymbol{\eta}^\top \mathbf{y}_{ad}), \quad (2.4)$$

where $\boldsymbol{\eta}$ is a Q -dimensional vector of coefficients with a Normal prior $\boldsymbol{\eta} \sim N(\boldsymbol{\mu}_\eta, \Sigma_\eta)$, \mathbf{y}_{ad} is a set of event timing features, e.g., any covariates that could affect timestamps of the edge, and $g(\cdot)$ is the appropriate link function such as identity, log, or inverse.

In modeling “when,” we do not directly model the timestamp t_d . Instead, we assume that each sender’s “time increment”—i.e., waiting time to next interaction—is drawn from a specific distribution in the exponential family. We define the time increment from edge $d - 1$ to edge d as τ_d (i.e., $\tau_d = t_d - t_{d-1}$) and specify the distribution of hypothetical timestamps with sender-specific mean μ_{ad} . Following the generalized linear model (GLM) framework (Nelder and Baker, 1972), we assume the mean and variance of the τ_{ad} satisfy

$$\begin{aligned} E(\tau_{ad}) &= \mu_{ad}, \\ V(\tau_{ad}) &= V(\mu_{ad}), \end{aligned} \quad (2.5)$$

where τ_{ad} here is a positive real number. Possible choices of distribution include exponential, Weibull, gamma, and log-normal² distributions, which are commonly used in time-to-event modeling (Rao, 2000; Rizopoulos, 2012). Based on the choice of distribution, we may introduce any additional latent variable (e.g., shape parameter k for Weibull, shape parameter θ for gamma, and variance parameter σ_τ^2 for log-normal) to account for the variance in time increments. We use $f_\tau(\cdot; \mu, V(\mu))$ and $F_\tau(\cdot; \mu, V(\mu))$ to denote the probability density function (p.d.f) and cumulative density function (c.d.f), respectively, with mean μ and variance $V(\mu)$.

2.3 Senders, Receivers, and Timestamps

Finally, we choose the actual sender, receivers, and timestamp by selecting the sender-receiver-set pair with the smallest time increment (Snijders, 1996):

$$\begin{aligned} a_d &= \operatorname{argmin}_a(\tau_{ad}), \\ \mathbf{r}_d &= \mathbf{u}_{a_d d}, \\ t_d &= t_{d-1} + \tau_{a_d d}. \end{aligned} \quad (2.6)$$

Therefore, it is a sender-driven process in that the receivers and timestamp of an edge is jointly determined by the sender’s urgency to send the edge to chosen receivers. Note that our generative process accounts for tied events such that in case of tied events (i.e., multiple senders generated exactly same time increments), we observe all of the tied events without assigning the orders of tied events. Algorithm 1 summarizes the entire generative process for directed edges with one sender and one or more receivers, outlined in this section.

²The log-normal distribution is not exponential family but can be used via modeling of $\log(\tau_d)$.

Algorithm 1 Generative Process: one sender and one or more receivers

Input: number of edges and nodes (D, A), covariates (\mathbf{x}, \mathbf{y}), and the coefficients ($\mathbf{b}, \boldsymbol{\eta}$)

```

for d=1 to D do
    for a=1 to A do
        for r=1 to A ( $r \neq a$ ) do
            set  $\lambda_{adr} = \mathbf{b}^\top \mathbf{x}_{adr}$ 
        end for
        draw  $\mathbf{u}_{ad} \sim \text{MB}_G(\boldsymbol{\lambda}_{ad})$ 
        set  $\mu_{ad} = g^{-1}(\boldsymbol{\eta}^\top \mathbf{y}_{ad})$ 
        draw  $\tau_{ad} \sim f_\tau(\mu_{ad}, V(\mu_{ad}))$ 
    end for
    if  $n \geq 2$  tied events then
        set  $a_d, \dots, a_{d+n-1} = \text{argmin}_a(\tau_{ad})$ 
        set  $\mathbf{r}_d = \mathbf{u}_{add}, \dots, \mathbf{r}_{d+n-1} = \mathbf{u}_{a_{d+n-1}d}$ 
        set  $t_d, \dots, t_{d+n-1} = t_{d-1} + \min_a \tau_{ad}$ 
        jump to  $d = d + n$ 
    else
        set  $a_d = \text{argmin}_a(\tau_{ad})$ 
        set  $\mathbf{r}_d = \mathbf{u}_{add}$ 
        set  $t_d = t_{d-1} + \min_a \tau_{ad}$ 
    end if
end for

```

3 Posterior Inference

Our inference goal is to invert the generative process to obtain the posterior distribution over the latent variables—hypothetical edges $\{\mathbf{u}_d\}_{d=1}^D$, coefficients for edge covariates \mathbf{b} , and coefficients for timestamp covariates $\boldsymbol{\eta}$ —conditioned on the observed data $\{(a_d, \mathbf{r}_d, t_d)\}_{d=1}^D$, covariates $\{(\mathbf{x}_d, \mathbf{y}_d)\}_{d=1}^D$, and hyperparameters $(\boldsymbol{\mu}_b, \Sigma_b, \boldsymbol{\mu}_\eta, \Sigma_\eta)$. We draw the samples using Markov chain Monte Carlo (MCMC) methods, repeatedly resampling the value of each latent variable from its conditional posterior via a Metropolis-within-Gibbs sampling algorithm. In this section, we provide each latent variable’s conditional posterior, and demonstrate how we perform the prior-posterior simulator test of [Geweke \(2004\)](#) for the HEM. At the end of this section, we provide the pseudocode of our MCMC algorithm in Algorithm 2.

3.1 Conditional Posteriors

Hypothetical edges

In the HEM, direct computation of the posterior densities for the latent variables \mathbf{b} and $\boldsymbol{\eta}$ —i.e., $P(\mathbf{b}|\mathbf{x}, \mathbf{a}, \mathbf{r}, \mathbf{t})$ and $P(\boldsymbol{\eta}|\mathbf{y}, \mathbf{a}, \mathbf{r}, \mathbf{t})$ —are not possible. However, it is possible to augment the data by hypothetical edges \mathbf{u} such that we can obtain their conditional posterior by collapsing the known distributions— $P(\mathbf{b}, \mathbf{u}|\mathbf{x}, \mathbf{a}, \mathbf{r}, \mathbf{t})$ and $P(\boldsymbol{\eta}, \mathbf{u}|\mathbf{y}, \mathbf{a}, \mathbf{r}, \mathbf{t})$ —

through integrating out \mathbf{u} . We adapt this common tool in Bayesian statistics called “data augmentation” (Tanner and Wong, 1987; Neal and Kypraios, 2015). Since u_{adr} is a binary random variable, its new value may be sampled directly from a multinomial distribution with probabilities

$$\begin{aligned}\Pr(u_{adr} = 1 | \mathbf{u}_{ad \setminus r}, \mathbf{b}, \mathbf{x}, \mathbf{a}, \mathbf{r}, \mathbf{t}) &\propto \exp(\lambda_{adr}); \\ \Pr(u_{adr} = 0 | \mathbf{u}_{ad \setminus r}, \mathbf{b}, \mathbf{x}, \mathbf{a}, \mathbf{r}, \mathbf{t}) &\propto I(\|\mathbf{u}_{ad \setminus r}\|_1 > 0),\end{aligned}\quad (3.1)$$

where $I(\cdot)$ is the indicator function that is used to prevent from the instances where a sender chooses zero number of receivers.

Coefficients for edge covariates

Unlike hypothetical edges above, new values for \mathbf{b} cannot be sampled directly from its conditional posterior, but may instead be obtained using the Metropolis–Hastings (M-H) algorithm. Assuming an uninformative prior (i.e., $N(0, \infty)$), the conditional posterior over \mathbf{b} is

$$\Pr(\mathbf{b} | \mathbf{u}, \mathbf{x}, \mathbf{a}, \mathbf{r}, \mathbf{t}) \propto \prod_{d=1}^D \prod_{a=1}^A \frac{1}{Z(\boldsymbol{\lambda}_{ad})} \exp \left(\log(I(\|\mathbf{u}_{ad}\|_1 > 0)) + \sum_{r \neq a} \lambda_{adr} u_{adr} \right). \quad (3.2)$$

Coefficients for timestamp covariates

Likewise, we use the M-H algorithm to update the latent variable $\boldsymbol{\eta}$. Assuming an uninformative prior $\boldsymbol{\eta}$ (i.e., $N(0, \infty)$), the conditional posterior for no-tied event case is

$$\Pr(\boldsymbol{\eta} | \mathbf{u}, \mathbf{y}, \mathbf{a}, \mathbf{r}, \mathbf{t}) \propto \prod_{d=1}^D \left(f_\tau(\tau_d; \mu_{ad}, V(\mu_{ad})) \times \prod_{a \neq ad} (1 - F_\tau(\tau_d; \mu_{ad}, V(\mu_{ad}))) \right), \quad (3.3)$$

where $f_\tau(\tau_d; \mu_{ad}, V(\mu_{ad}))$ is the probability that the d^{th} observed time increment comes from the specified distribution $f_\tau(\cdot)$ with the observed sender’s mean μ_{ad} , and $\prod_{a \neq ad} (1 - F_\tau(\tau_d; \mu_{ad}, V(\mu_{ad})))$ is the probability that the rest of (unobserved) senders for event d all generated time increments greater than τ_d . Moreover, under the existence of tied-event, the conditional posterior of $\boldsymbol{\eta}$ is written as

$$\begin{aligned}\Pr(\boldsymbol{\eta} | \mathbf{u}, \mathbf{y}, \mathbf{a}, \mathbf{r}, \mathbf{t}) &\propto \prod_{m=1}^M \left(\prod_{d: t_d = t_m^*} f_\tau(t_m^* - t_{m-1}^*; \mu_{ad}, V(\mu_{ad})) \right. \\ &\quad \times \left. \prod_{a \notin \{a_d\}_{d: t_d = t_m^*}} (1 - F_\tau(t_m^* - t_{m-1}^*; \mu_{ad}, V(\mu_{ad}))) \right),\end{aligned}\quad (3.4)$$

where t_1^*, \dots, t_M^* are the unique timepoints across D events ($M \leq D$). If $M = D$ (i.e., no tied events), equation (3.4) reduces to equation (3.3). Note that when we have the latent variable to quantify the variance in time increments $V(\mu)$ (based on the choice of timestamp distribution in Section 2.2), we also use equation (3.3) (or equation (3.4) in

Algorithm 2 MCMC Algorithm

Input: number of outer and inner iterations (O, I_1, I_2) and initial values of $(\mathbf{u}, \mathbf{b}, \boldsymbol{\eta})$

```

for o=1 to O do
    for d=1 to D do
        for a = 1 to A do
            for r = 1 to A ( $r \neq a$ ) do
                update  $u_{adr}$  using Gibbs update—equation (3.1)
            end for
        end for
    end for
    for n=1 to  $I_1$  do
        update  $\mathbf{b}$  using M-H algorithm—equation (3.2)
    end for
    for n=1 to  $I_2$  do
        update  $\boldsymbol{\eta}$  using M-H algorithm—equation (3.3) or (3.4)
    end for
    if extra parameter for  $V(\mu)$  then
        update the variance parameter using M-H algorithm—equation (3.3) or (3.4)
    end if
end for
summarize the results with the last chain of  $\mathbf{b}$  and  $\boldsymbol{\eta}$ 

```

case there exist tied events) for the additional M-H update—e.g., $\Pr(k|\boldsymbol{\eta}, \mathbf{u}, \mathbf{y}, \mathbf{a}, \mathbf{r}, \mathbf{t})$ for Weibull, $\Pr(\theta|\boldsymbol{\eta}, \mathbf{u}, \mathbf{y}, \mathbf{a}, \mathbf{r}, \mathbf{t})$ for gamma, and $\Pr(\sigma_\tau^2|\boldsymbol{\eta}, \mathbf{u}, \mathbf{y}, \mathbf{a}, \mathbf{r}, \mathbf{t})$ for log-normal distribution.

3.2 Getting It Right (GiR) Test

Software development is integral to the objective of applying the HEM to real world data. Code review is a valuable process in any research computing context, and the prevalence of software bugs in statistical software is well documented (e.g., Altman et al., 2004; McCullough, 2009). With highly complex models such as the HEM, there are many ways in which software bugs can be introduced and go unnoticed. As such, we present a joint analysis of the integrity of our generative model, sampling equations, and software implementation.

Geweke (2004) introduced the “Getting it Right” (GiR) test—a joint distribution test of posterior simulators which can detect errors in sampling equations as well as coding errors—and it has been used to test the implementation of Bayesian inference algorithms (Zhao et al., 2016). The test involves comparing the distributions of variables simulated from two joint distribution samplers, which we call “forward” and “backward” samples. The forward sampler draws unobservable variables from the prior and then generates the observable data conditional on the unobservables. The backward sampler alternates between the inference and an observables simulator, by running the inference

code on observable data to obtain posterior estimates of the unobservable variables and then re-generating the observables given the inferred unobservables. The backward sampler is initialized by running an iteration of inference on observables drawn directly from the prior. Since the only information on which both the forward and backward samplers are based is the prior, if the sampling equations are correct and the code is implemented without bugs, each variable should have the same distribution in the forward and backward samples.

In the forward samples, both observable and unobservable variables are generated using Algorithm 1. In the backward samples, unobservable variables are generated using the sampling equations for inference, which we derived in Section 3.1. For each forward and backward sample that consists of D number of edges, we save these statistics:

1. Mean of observed receiver sizes $\|\mathbf{r}_d\|_1$ across $d = 1, \dots, D$,
2. Variance of observed receiver sizes $\|\mathbf{r}_d\|_1$ across $d = 1, \dots, D$,
3. Mean of time increments τ_d across $d = 1, \dots, D$,
4. Variance of time increments τ_d across $d = 1, \dots, D$,
5. b_p value used to generate the samples $p = 1, \dots, P$,
6. η_q value used to generate the samples $q = 1, \dots, Q$,
7. σ_τ^2 value used to generate the samples in log-normal distribution

To keep the computational burden of re-running thousands of rounds of inference manageable, we run the GiR using a relatively small artificial sample, consisting of $D = 100$ edges, $A = 5$ actors, $P = 4$ number of edge covariates, and $Q = 3$ number of timestamp covariates per each forward or backward samples, using log-normal distribution for the time increments f_τ . We generated 10^5 sets of forward and backward samples, and then calculated 1,000 quantiles for each of the statistics. We also calculated t-test and Mann-Whitney test p-values in order to test for differences in the distributions generated in the forward and backward samples. Before we calculated these statistics, we thinned our samples by taking every 9th sample starting at the 10,000th sample for a resulting sample size of 10,000, in order to reduce the autocorrelation in the Markov chains. In each case, if we observe a large p-value, this gives us evidence that the distributions generated under forward and backward sampling have the same locations. We depict the GiR results using probability-probability (PP) plots. To compare two samples with PP plots we calculate the empirical quantile in each sample of a set of values observed across the two samples, then plot the sets of quantiles in the two samples against each other. If the two samples are from equivalent distributions, the quantiles should line up on a line with zero y -intercept, and unit slope (i.e., a 45-degree line). The GiR test results are depicted in Figure 1, which show that we pass the test on every statistic.

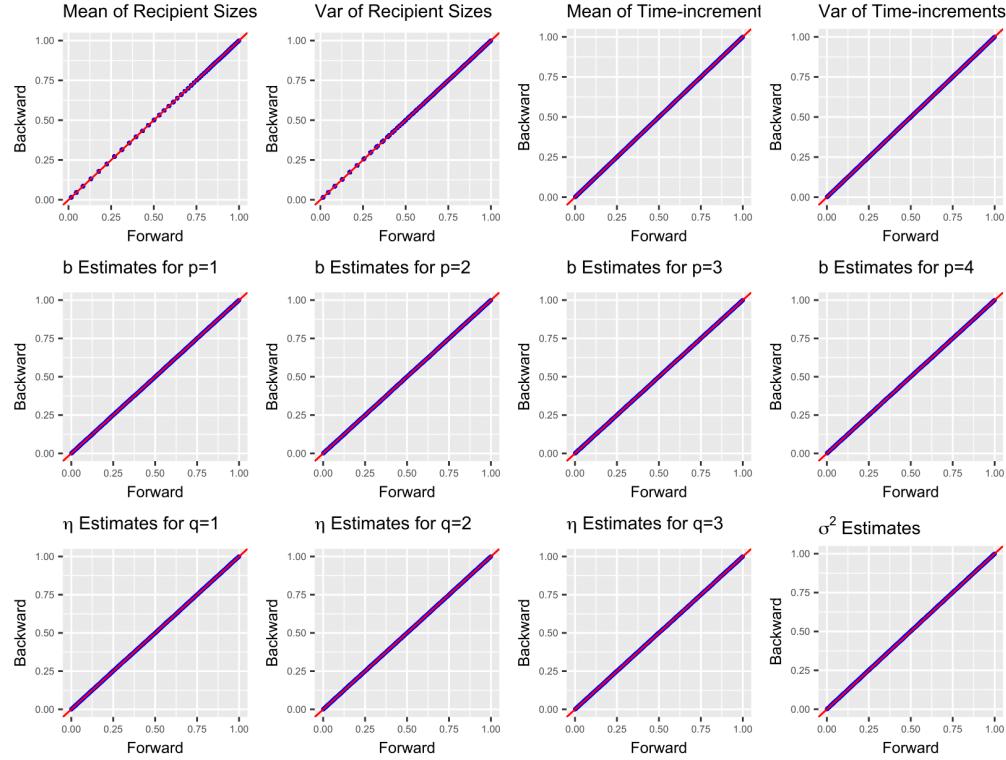


Figure 1: Probability-probability (PP) plots for the GiR test statistics.

4 Application to Email Data

We now present a case study applying our method to Montgomery county government email data. Our data come from the North Carolina county government email dataset collected by [ben Aaron et al. \(2017\)](#) that includes internal email corpora covering the inboxes and outboxes of managerial-level employees of North Carolina county governments. Out of over twenty counties, we chose Montgomery County to 1) test our model using data with large proportion of hyperedges (16.76%), all of which are emails sent from one sender to two or more receivers, and 2) limit the scope of this initial application. To summarize, Montgomery County email network contains 680 emails, sent and received by 18 department managers over a period of 3 months (March–May) in 2012. For this case study, we formulate the edge covariates \mathbf{x} and timestamp covariates \mathbf{y} and ground them in illustrative examples. We then report a suite of experiments that test the HEM’s ability to form the posterior distribution over its latent variables.

4.1 Covariates

Edge covariates

Given a set of network data, a primary modeling goal of the generative process for edges lies in determining which characteristics and behaviors are predictive of the receiver selection. This email application specifically give rise to the following question: “To what extent are nodal, dyadic or triadic network effects relevant to predicting future emails?” As an illustrative example, we form the receiver selection features \mathbf{x} for Montgomery County email data using some nodal statistics and interval-based network statistics. First, we include the gender information of sender and receiver, as well as their homophily indicator. For dyadic and triadic network effects, we summarize past interaction behaviors based on the time interval prior to and including t_{d-1} . Specifically, our time interval tracks 7 days prior to the last email was sent $l_d = (t_{d-1} - 7 \text{ days}, t_{d-1}]$. For $a \in [A]$, $r \in [A]$, and $d \in [D]$, we define 14 covariates for \mathbf{x}_{adr} :

1. intercept: $x_{adr1} = 1$;
2. gender of sender: $x_{adr2} = I(\text{gender of sender } a = \text{female})$;
3. gender of receiver: $x_{adr3} = I(\text{gender of receiver } r = \text{female})$;
4. gender homophily: $x_{adr4} = I(x_{adr2} = x_{adr3})$;
5. outdegree of sender: $x_{adr5} = \sum_{d':t_{d'} \in l_d} I(a_{d'} = a)$;
6. indegree of receiver: $x_{adr6} = \sum_{d':t_{d'} \in l_d} I(u_{d'r} = 1)$;
7. hyperedge size of sender: $x_{adr7} = \sum_{d':t_{d'} \in l_d} \sum_{r=1}^A I(a_{d'} = a) I(u_{d'r} = 1)$;
8. interaction between outdegree and hyperedge size: $x_{adr8} = x_{adr5} \times x_{adr7}$;
9. send: $x_{adr9} = \sum_{d':t_{d'} \in l_d} I(a_{d'} = a) I(u_{d'r} = 1)$;
10. receive: $x_{adr10} = \text{send}(r, a)$;
11. 2-send: $x_{adr11} = \sum_{h \neq a, r} \text{send}(a, h) \text{send}(h, r)$;
12. 2-receive: $x_{adr12} = \sum_{h \neq a, r} \text{send}(h, a) \text{send}(r, h)$;
13. sibling: $x_{adr13} = \sum_{h \neq a, r} \text{send}(h, a) \text{send}(h, r)$;
14. cosibling: $x_{adr14} = \sum_{h \neq a, r} \text{send}(a, h) \text{send}(r, h)$;

where $I(\cdot)$ is an indicator function. The network statistics (5–14) are designed so that their coefficient have a straightforward interpretation. The statistics “outdegree” and “indegree” measures the gregariousness and popularity effect of the node by counting the number of emails sent from a and received by r , respectively, within the last 7 days. Moreover, in order to capture individual tendency of send emails to two or more receivers, we include the statistic “hyperedge size”—the number of emails sent from a

$$\begin{array}{ll}
 \textbf{2-send} & \sum_h (a \rightarrow h \rightarrow r) \\
 \textbf{2-receive} & \sum_h (a \leftarrow h \leftarrow r) \\
 \textbf{sibling} & \sum_h (h \xleftrightarrow{a} r) \\
 \textbf{cosibling} & \sum_h (h \xleftarrow{a} r)
 \end{array}$$

Figure 2: Visualization of triadic statistics: 2-send, 2-receive, sibling, and cosibling.

within last 7 days where emails with k number of receivers are counted as k separate emails—as a variant of outdegree statistic, accounting for hyperedges. We also include the interaction term between outdegree and hyperedge size. Next, we employ dyadic and triadic network statistics from Perry and Wolfe (2013). Dyadic statistics “send” and “receive” are defined as above such that these covariates measure the number of emails sent from a to r and r to a , respectively, within the last 7 days. In the example of triadic statistics, the covariate “2-send” counts the pairs of emails involving some node h distinct from a and r such that emails from a to h and h to r are both observed within the last 7 days. Other triadic covariates behave similarly, and their interpretations are also analogous, which are illustrated in Figure 2.

Timestamp covariates

For the event timing features \mathbf{y} introduced in Section 2.2, we identify a set of covariates which may possibly effect “time to the next email.” Similar to the edge covariates, we include nodal statistics which are time-invariant (such as gender or manager status) or time-dependent (such as the network statistics used for \mathbf{x}). In addition, we select some edge-specific covariates based on the temporal aspect of the $(d-1)^{th}$ email—e.g., whether the previous email was sent (1) during the weekend and (2) before or past midday (AM/PM)—since we expect the email interactions within county government to be less active during the weekend and in the evening. To be specific, the timestamp statistics are defined as

1. intercept: $y_{ad1} = 1$;
2. gender of sender: $y_{ad2} = I(\text{gender of sender } a = \text{female})$;
3. manager status of sender: $y_{ad3} = I(\text{sender } a \text{ is the County Manager})$;
4. outdegree of sender: $y_{ad4} = \sum_{d':t_{d'} \in l_d} I(a_{d'} = a)$;
5. indegree of sender: $y_{ad5} = \sum_{d':t_{d'} \in l_d} I(u_{d'a} = 1)$;
6. weekend indicator of previous email: $y_{ad6} = I(t_{d-1} \text{ is during the weekend})$;
7. AM/PM of previous email: $y_{ad7} = I(t_{d-1} \text{ in PM})$.

Note that our generative process for timestamps in Section 2.2 is sender-oriented process where the sender determines “when to send the email,” thus we incorporate network

statistics that depends on a only—outdegree of sender a and indegree of sender a (not r this time).

4.2 Model Selection

The HEM has many component parts that need to be specified by the user (i.e., the receiver selection features \mathbf{x} , the selection of the event timing features \mathbf{y} , and the distribution of time increments f). Many of these components will be specified based on user expertise (e.g., regarding which features would drive receiver selection), but some decisions may require a data-driven approach to model specification. For example, though theoretical considerations may inform the specification of features, subject-matter expertise is unlikely to inform the decision regarding the family of the event time distribution. Furthermore, since different distribution families (and model specifications more generally) may involve different size parameter spaces, any data-driven approach to model comparison must guard against over-fitting the data. In this section we present a general-purpose approach to evaluating the HEM specification using out-of-sample prediction. We illustrate this approach by comparing alternative distributional families for the event timing component of the model. Here, we specifically compare the predictive performance from two distributions—log-normal and exponential. We particularly choose the log-normal distribution based on some exploratory analysis (e.g., histogram and simple regressions) on raw time increments data, and take exponential distribution as an alternative since exponential is the most commonly specified distribution for time-to-event data which is also used in the stochastic actor-oriented models (SAOMs) ([Snijders, 1996](#)) as well as their extensions ([Snijders et al., 2007](#)).

We evaluate the model’s ability to predict edges and timestamps from Montgomery County email data, conditioned on their “training” part of the data. To perform the experiment, we separately formed a test split of each three model components—sender, receivers, and timestamps—by randomly selecting “test” data with probability $p = 0.1$. Any missing variables were imputed by drawing samples from their conditional posterior distributions, given the observed data, model estimates, and current values of test data. We then run inference to update the latent variables given the imputed and observed data. We iterate the two steps—imputation and inference—multiple times to obtain enough number of estimates for test data. Algorithm 3 outlines this procedure in detail.

We run the experiment and measure the predictive performance of two separate time distributions using $N = 500$ predicted samples, by comparing the predictions in terms of classification accuracy in predicting the senders and receivers, as well as prediction error in the timestamps. First, we compare the posterior probability of correct senders for each of missing emails $\{d : a_d = \text{NA}\}$, which corresponds to $\pi_{a_d} = P(a_d = a_d^{obs} | \cdot)$ in Algorithm 3. We call this measure as “correct sender posterior probability.” On the left of Figure 3, we draw boxplots for the distribution of mean correct sender posterior probability—i.e., $\hat{\pi}_{a_d} = \frac{1}{N} \sum_{n=1}^N \pi_{a_d}^n$ —across the missing emails. The results show that both log-normal and exponential distributions achieve better predictive performance for missing senders compared to what is expected under random guess (i.e., choose one out of A possible senders = 1/18), with log-normal distribution showing slightly higher per-

Algorithm 3 Out-of-Sample Predictions

Input: data $\{(a_d, \mathbf{r}_d, t_d)\}_{d=1}^D$, number of new data to generate R , and initial values of $(\mathbf{b}, \boldsymbol{\eta}, \mathbf{u}, \sigma_\tau^2)$

Test splits (with $p = 0.1$):
draw test senders (out of D senders)
draw test receivers (out of $D \times (A - 1)$ receiver indicators $\{\{\mathbf{r}_{dr}\}_{r \in [A] \setminus a_d}\}_{d=1}^D$)
draw test timestamps (out of D timestamps)
set the test data as “missing” (NA)

Imputation and inference:

```

for  $r = 1$  to  $R$  do
    for  $d = 1$  to  $D$  do
        if  $a_d = \text{NA}$  then
            for  $a = 1$  to  $A$  do
                compute  $\pi_a = P(a_d = a | \cdot)$  using equation (3.3) (without the product term)
            end for
            draw  $a_d \sim \text{multinomial}(\pi_a)$ 
        end if
        for  $r \in [A] \setminus a_d$  do
            if  $r_{dr} = \text{NA}$  then
                draw  $r_{dr}$  from multinomial with probability  $P(r_{dr} = 1 | \cdot)$  and  $P(r_{dr} = 0 | \cdot)$ 
                using equation (3.1)
            end if
        end for
        if  $t_d = \text{NA}$  then
            draw  $\tau_d^{new}$  from equation (3.3) (without the product term) via importance sampling3
        end if
        run inference and update  $(\mathbf{u}, \mathbf{b}, \boldsymbol{\eta})$  given the imputed and observed data
    end for
    store the estimates for test data
end for

```

formance than exponential distribution on average. Secondly, since the receiver vector is binary, we compute F_1 scores for missing receiver indicators (i.e., all d and r with $r_{dr}=\text{NA}$) by taking the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \text{ where} \\ \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \text{ and precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (4.1)$$

with TP denoting true positive (i.e., $\mathbf{r}_{dr}^{obs} = \mathbf{r}_{dr}^{pred} = 1$), FN denoting false negative (i.e., $\mathbf{r}_{dr}^{obs} = 1$ but $\mathbf{r}_{dr}^{pred} = 0$), and FP denoting false positive (i.e., $\mathbf{r}_{dr}^{obs} = 0$ but $\mathbf{r}_{dr}^{pred} = 1$). Although the generative process for edges (Section 2.1) is not directly affected by the

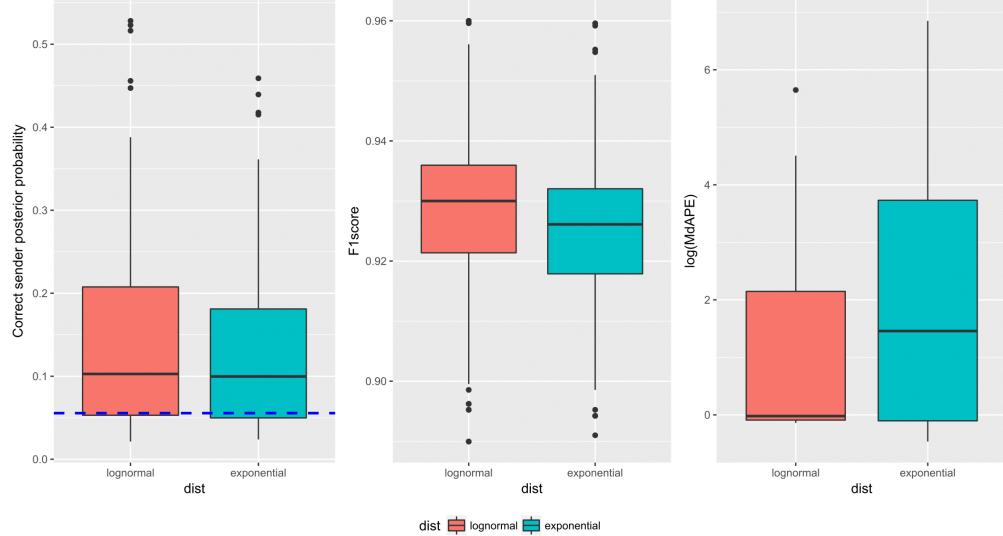


Figure 3: Comparison of predictive performance between log-normal and exponential distributions: boxplots for correct sender posterior probability (*left*), F_1 scores from receiver predictions (*middle*), and boxplots for median absolute percentage error (*right*). The dotted blue lines in the left plot represents the correct sender probability expected by random guess—i.e., $1/A = 1/18 \approx 0.056$.

choice of timestamp distribution, the middle plot in Figure 3 reveals noticeable difference between log-normal and exponential in their performance in predicting missing receiver indicators, where log-normal outperforms exponential. Finally, the prediction error for d^{th} missing timestamp is estimated using the median of absolute relative errors⁴ across $N = 500$ predictions:

$$e_{\tau_d} = \text{median}\left(\left|\frac{\tau_d^{obs} - \tau_d^{pred_1}}{\tau_d^{obs}}\right|, \dots, \left|\frac{\tau_d^{obs} - \tau_d^{pred_N}}{\tau_d^{obs}}\right|\right). \quad (4.2)$$

The right plot of Figure 3 presents boxplots for the median absolute relative errors, where plot the estimates in a log-scale. Surprisingly, we have a huge benefit in the performance of timestamp prediction when we assume log-normal distribution for time-increments compared to exponential distribution. This difference can be explained by overdispersion in exponential distribution, because there exists greater variability in the time increments of emails than would be expected under exponential distribution. As illustrated above, we can use this out-of-sample prediction task for two usage—1) provide an effective answer to the question “how are we filling in the missing components of time-stamped network data?” and 2) offer one standard way to determine the distribution of time increments in Section 2.2.

⁴Hyndman and Koehler (2006) refer to this as median absolute percentage error (MdAPE).

4.3 Posterior Predictive Checks

In this section, we perform posterior predictive checks (PPC) (Rubin et al., 1984) to evaluate the appropriateness of our model specification for Montgomery County email data. We formally generated entirely new data, by simulating $N = 500$ synthetic email data $\{(a_d, \mathbf{r}_d, t_d)\}_{d=1}^D$ from the generative process in Section 2, conditional upon a set of inferred latent variables from the inference in Section 4.4. For the test of goodness-of-fit in terms of network dynamics, we use multiple statistics that summarize meaningful aspects of the data: outdegree distribution, indegree distribution, receiver size distribution, and probability-probability (PP) plot for time increments.

Figure 4 illustrates the results of posterior predictive checks using log-normal distribution, which shows better performance in Section 4.2. Upper two plots show node-specific posterior predictive degree distributions across $N = 500$ synthetic data, where the left one for outdegree statistic and the right plot is for indegree statistic. For both plots, the x-axis represents the nodes ($A = 1, \dots, 18$), and the y-axis represents the number of emails sent or received by the node. When compared with the observed outdegree and indegree statistics (red lines), our model recovers the overall distribution of sending and receiving activities across the nodes. For example, node 1 and 10 show significantly higher level of both sending and receiving activities relative to the rest, and the model-simulated data captures those big jumps, showing acceptable fit to the data. Outdegree distribution of some low-activity nodes are not precisely recovered, however, indegree distribution looks much better. Since we use more information in the receiver selection process (i.e., network effects) while we rely solely on minimum time increments when choosing the observed sender, these results are expected. Lower left plot is the distribution of receiver sizes, where the x-axis spans over the size of receivers 1 to 14 (which is the maximum size of observed receivers) and the y-axis denotes the number of emails with x-number of receivers. The result shows that our model is underestimating emails with one receiver while overestimating emails with two, three, and four receivers. One explanation behind what we observe is that the model is trying to recover so-called “broadcast” emails, which are the emails with ≥ 10 number of receivers, so that the intercept estimate b_1 is slightly moved toward right. To improve the goodness of fit for receiver sizes, we can add more covariates (e.g., sender indicators) or assign strong prior structure on b_1 . In the end, the plot on the lower right is the PP plot for time increments, which depicts the two cumulative distribution functions against each other in order to assess how closely two data sets agree. Here, the closeness to the diagonal line gives a measure of the difference between the distribution of simulated time increments and the observed ones, and our PP plot shows that we have great performance in reproducing the observed time distribution. All our findings from predictive experiments in Section 4.2 are further revealed in the PPC from exponential distribution, where the comparison between log-normal and exponential distributions are attached in Appendix C.

4.4 Exploratory Analysis

Based on the prediction experiments in Section 4.2, we choose to analyze Montgomery county email data and interpret the results using log-normal distribution. We assume

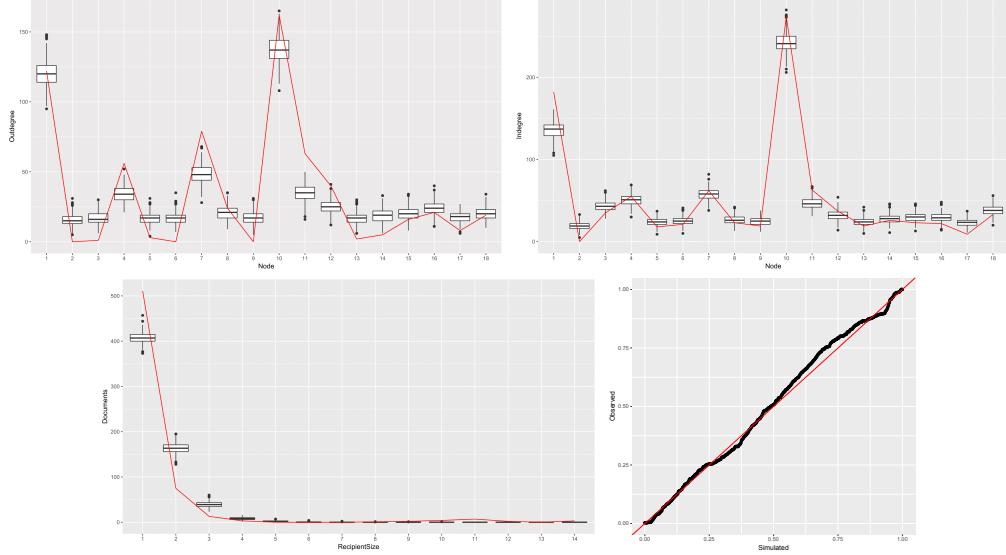


Figure 4: PPC results from log-normal distribution: outdegree distribution (*upper left*), indegree distribution (*upper right*), receiver size distribution (*lower left*), and PP plot for time increments (*lower right*). Blue lines denote the observed statistics (in the first three plots) or the diagonal line connecting $(0, 0)$ and $(1, 1)$ (in the last plot).

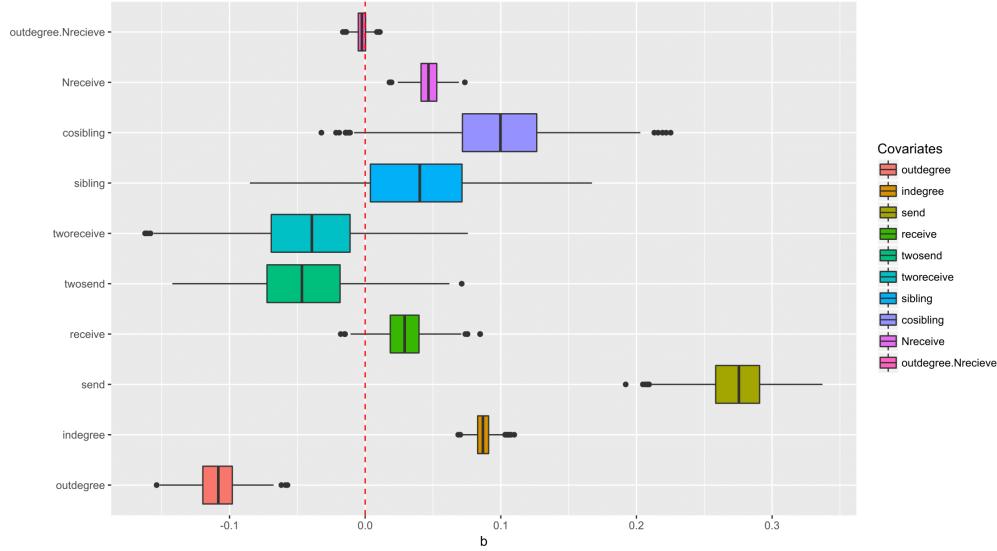
weakly informative priors for latent variables such as $\mathbf{b} \sim N(\boldsymbol{\mu}_b = \mathbf{0}, \Sigma_b = 2 \times I_P)$, $\boldsymbol{\eta} \sim N(\boldsymbol{\mu}_\eta = \mathbf{0}, \Sigma_\eta = 2 \times I_Q)$, and $\sigma_\tau^2 \sim \text{inverse-Gamma}(a = 2, b = 1)$, and run MCMC algorithm in Algorithm 2 with $O = 55,000$ outer iterations with a burn-in of 15,000, where we thin by keeping every 40th sample. While the inner iterations for σ_τ^2 is fixed as 1, we specify the inner iterations $I_1 = 20$ for \mathbf{b} and $I_2 = 10$ for $\boldsymbol{\eta}$ to adjust for slower convergence rates. Convergence diagnostics including some traceplots and Geweke diagnostics (Geweke et al., 1991) are provided in Appendix D.

Coefficients for edge covariates

Figure 5 shows the boxplots summarizing posterior samples of \mathbf{b} . Since we use the logit probability of λ_{adr} when generating hypothetical edges in Section 2.1, we assume that

$$\text{logit}(\lambda_{adr}) = \log \left(\frac{\lambda_{adr}}{1 - \lambda_{adr}} \right) = b_1 + b_2 x_{adr2} \dots + b_{14} x_{adr14},$$

and can interpret the \mathbf{b} estimates in terms of odds ratio $\frac{\lambda_{adr}}{1 - \lambda_{adr}} = \exp(b_1 + b_2 x_{adr2} \dots + b_{14} x_{adr14})$. To begin with, we can see that the effect of the statistic “send” (i.e., number of times sender a sent emails to receiver r over the last week) is significant and positive with $\hat{b}_4 = 0.274$, implying that if sender a sent n number of emails to r last week, then sender a is approximately $\exp(0.274 \times n) \approx (1.315)^n$ times more likely to send an email to r . Similar interpretation can be applied for the covariates “indegree” and “hyperedge”

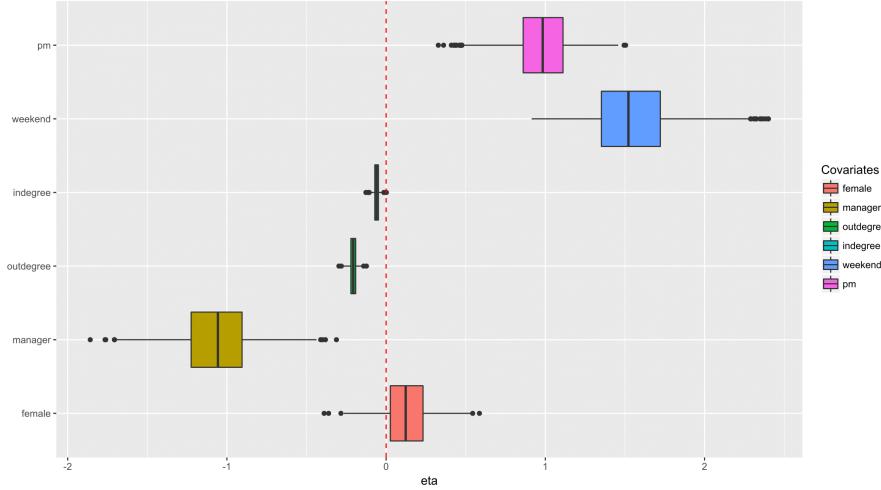
Figure 5: Posterior distribution of \boldsymbol{b} estimates.

size". For example, if the receiver r received n number of emails and sender a sent emails to n number of receivers over the last week, sender a is $\exp(0.086 \times n) \approx (1.091)^n$ times and $\exp(0.047 \times n) \approx (1.048)^n$ times, more likely to send an email to r . On the contrary, the statistic "outdegree" has significantly negative effect—i.e., if sender a sent n number of emails to anyone last week, then sender a is approximately $\exp(-0.109 \times n) \approx (0.897)^n$ times less likely to send an email to r —possibly due to its multicollinearity with the statistics "send" and/or "hyperedge size". This result is likely to appear under the following scenario: if sender a sent large number of emails over the last week but 1) not to r (excluding the effect of "send") and the email of interest is not a 2) broadcast email which is originally aimed at large number of receivers (excluding the effect of "hyperedge size"), then sender a is less likely to send an email to receiver r . Rest of statistics are not statistically significant in that their 95% credible interval do not include 0. Surprisingly, none of the covariates for triadic network effects seem to have significant effect on the generative process of edges.

Coefficients for timestamp covariates

For timestamp covariates, Figure 6 shows the boxplots summarizing posterior samples of $\boldsymbol{\eta}$. Since we assume log-normal distribution for time increments, the interpretation of $\hat{\boldsymbol{\eta}}$ should be based on

$$\log(\tau_{ad}) \sim N(\mu_{ad}, \sigma_\tau^2), \text{ with} \\ \mu_{ad} = \eta_1 + \eta_2 y_{ad2} + \dots + \eta_7 y_{ad7}.$$

Figure 6: Posterior distribution of η estimates.

To begin with, the posterior estimates of two temporal effects—weekend and pm—indicate that if the $(d - 1)^{th}$ email was sent during the weekend or after 12pm, then the time to d^{th} email is expected to take $\exp(1.552) \approx 4.722$ hours and $\exp(0.980) \approx 2.665$ hours longer, respectively, compared to their counterparts (i.e., weekdays and am). On the contrary, the statistics manager, outdegree, and indegree turn out to shorten the time to next email. For example, being a county manager (i.e., the manager of the managers) lowers the expected $\log(\tau_{ad})$ by -1.070, since he or she needs to respond fastly to emails with important decisions. In addition, the manager in general sends or receives a lot more emails which may shorten the response time. This argument is supported by the posterior estimates for outdegree and indegree, where the estimated coefficients are approximately -0.206 and -0.060, respectively. Gender of the department manager is shown to have no significant effect on time-to-next-email. The posterior mean estimate of the variance parameter σ_τ^2 is 14.093 with its 95% credible interval (12.709, 15.555), indicating that there exists huge variability in the time increments in this data.

5 Conclusion

Motivated by the relational event model (Butts, 2008) and its variants for time-stamped networks, the hyperedge event model (HEM) can effectively learn the underlying edge generating and timestamp generating structures of event data, while allowing flexibility in the covariate specifications and the distributional assumption of time increments. Accounting for the hyperedges, we make better use of events with hyperedges than modeling them as duplicate edges. In modeling the timestamps of the events, our generalized linear model (GLM) based formulation eliminates the need to stick with one distribution (e.g., exponential distribution), and we also provide an algorithm for pre-

dictive experiment that helps to learn which distribution better fits the data. To sum, the HEM's flexible generative process thus leads to more accurate and precise inference on model parameters.

We have demonstrated effectiveness of our model by analyzing Montgomery County government emails, which includes numerous hyperedges. The estimated covariate effects reveal that the HEM is able to understand the structural dynamics similar to those used in the exponential random graph model (ERGM), while the model additionally allows separate covariates and the corresponding parameters for generative processes of edges and timestamps. Although we illustrate the entire framework and application in the context of one type of hyperedges, a single sender and multiple receivers, our model can be easily extended to allow the opposite case, a single receiver and multiple senders, by slight modification of the generative process. This extension involves promising applications including international sanctions and co-sponsorship of bills. Furthermore, while currently the model's applicability is limited to small-sized networks since we assume fixed set of nodes across entire timepoints, we can adjust the model to allow time-varying nodes or improve its computational efficiency such that the model is well applicable to large-scale networks. Considering the recent explosion of network dataset with huge number of nodes (e.g., online communications), the development of model adjustments for better applicability are objects of future work.

Acknowledgments

This work was supported in part by the University of Massachusetts Amherst Center for Intelligent Information Retrieval and in part by National Science Foundation grants DGE-1144860, SES-1619644, and CISE-1320219. Any opinions, findings, and conclusions or recommendations are those of the authors and do not necessarily reflect those of the sponsors.

Appendix

Appendix A: Alternative Generative Process

Appendix B: Normalizing Constant of MB_G

Our probability measure “ MB_G ”—the multivariate Bernoulli distribution with non-empty Gibbs measure—defines the probability of sender a selecting the binary receiver vector \mathbf{u}_{ad} as

$$\Pr(\mathbf{u}_{ad}|\mathbf{b}, \mathbf{x}_{ad}) = \frac{1}{Z(\boldsymbol{\lambda}_{ad})} \exp \left(\log(I(\|\mathbf{u}_{ad}\|_1 > 0)) + \sum_{r \neq a} \lambda_{adr} u_{adr} \right),$$

where the receiver intensity is a linear combination of edge covariates—i.e., $\lambda_{adr} = \mathbf{b}^\top \mathbf{x}_{adr}$ —as defined in Section 2.1.

To use this distribution efficiently, we derive a closed-form expression for $Z(\boldsymbol{\lambda}_{ad})$ that does not require brute-force summation over the support of \mathbf{u}_{ad} (*i.e.* $\forall \mathbf{u}_{ad} \in [0, 1]^A$).

Algorithm 4 Generative Process: one receiver and one or more senders

Input: number of edges and nodes (D, A), covariates (\mathbf{x}, \mathbf{y}), and the coefficients ($\mathbf{b}, \boldsymbol{\eta}$)

```

for d=1 to D do
  for r=1 to A do
    for a=1 to A ( $a \neq r$ ) do
      set  $\lambda_{adr} = \mathbf{b}^\top \mathbf{x}_{adr}$ 
    end for
    draw  $\mathbf{u}_{rd} \sim \text{MB}_G(\boldsymbol{\lambda}_{rd})$ 
    set  $\mu_{rd} = g^{-1}(\boldsymbol{\eta}^\top \mathbf{y}_{rd})$ 
    draw  $\tau_{rd} \sim f_\tau(\mu_{rd}, \sigma_\tau^2)$ 
  end for
  if  $n \geq 2$  tied events then
    set  $r_d = \text{argmin}_r(\tau_{rd})$ 
    set  $\mathbf{a}_d = \mathbf{u}_{r_dd}, \dots, \mathbf{a}_{d+n-1} = \mathbf{u}_{r_{d+n-1}d}$ 
    set  $t_d, \dots, t_{d+n-1} = t_{d-1} + \min_r \tau_{rd}$ 
    jump to  $d = d + n$ 
  else
    set  $r_d = \text{argmin}_r(\tau_{rd})$ 
    set  $\mathbf{a}_d = \mathbf{u}_{r_dd}$ 
    set  $t_d = t_{d-1} + \min_r \tau_{rd}$ 
  end if
end for

```

We recognize that if \mathbf{u}_{ad} were drawn via independent Bernoulli distributions in which $\Pr(u_{adr} = 1 | \mathbf{b}, \mathbf{x}_{ad})$ was given by $\text{logit}(\lambda_{adr})$, then

$$\Pr(\mathbf{u}_{ad} | \mathbf{b}, \mathbf{x}_{ad}) \propto \exp\left(\sum_{r \neq a} \lambda_{adr} u_{adr}\right).$$

This is straightforward to verify by looking at

$$\Pr(u_{adr} = 1 | \mathbf{u}_{ad[-r]}, \mathbf{b}, \mathbf{x}_{ad}) = \frac{\exp(\lambda_{adr})}{\exp(\lambda_{adr}) + 1}.$$

We denote the logistic-Bernoulli normalizing constant as $Z^l(\boldsymbol{\lambda}_{ad})$, which is defined as

$$Z^l(\boldsymbol{\lambda}_{ad}) = \sum_{\mathbf{u}_{ad} \in [0,1]^A} \exp\left(\sum_{r \neq a} \lambda_{adr} u_{adr}\right).$$

Now, since

$$\exp\left(\log\left(\mathbb{I}(\|\mathbf{u}_{ad}\|_1 > 0)\right) + \sum_{r \neq a} \lambda_{adr} u_{adr}\right) = \exp\left(\sum_{r \neq a} \lambda_{adr} u_{adr}\right),$$

except when $\|\mathbf{u}_{ad}\|_1 = 0$, we note that

$$\begin{aligned} Z(\boldsymbol{\lambda}_{ad}) &= Z^l(\boldsymbol{\lambda}_{ad}) - \exp\left(\sum_{\forall u_{adr}=0} \lambda_{adr} u_{adr}\right) \\ &= Z^l(\boldsymbol{\lambda}_{ad}) - 1. \end{aligned}$$

We can therefore derive a closed form expression for $Z(\boldsymbol{\lambda}_{ad})$ via a closed form expression for $Z^l(\boldsymbol{\lambda}_{ad})$. This can be done by looking at the probability of the zero vector under the logistic-Bernoulli model:

$$\frac{1}{Z^l(\boldsymbol{\lambda}_{ad})} \exp\left(\sum_{\forall u_{adr}=0} \lambda_{adr} u_{adr}\right) = \prod_{r \neq a} \left(1 - \frac{\exp(\lambda_{adr})}{\exp(\lambda_{adr}) + 1}\right).$$

Then, we have

$$\frac{1}{Z^l(\boldsymbol{\lambda}_{ad})} = \prod_{r \neq a} \frac{1}{\exp(\lambda_{adr}) + 1}.$$

Finally, the closed form expression for the normalizing constant is

$$Z(\boldsymbol{\lambda}_{ad}) = \prod_{r \neq a} (\exp(\lambda_{adr}) + 1) - 1.$$

Appendix C: Comparison of PPC Results: log-normal vs. exponential

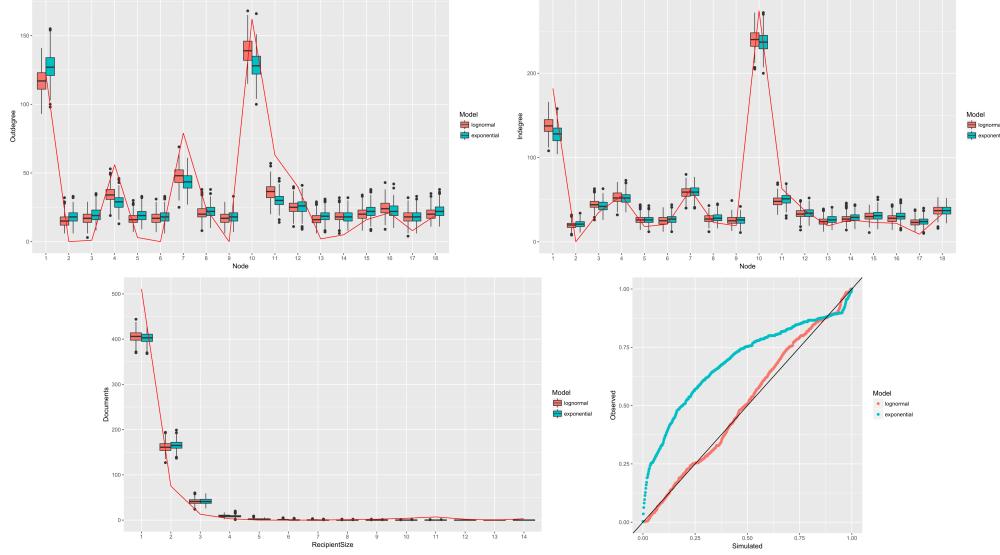


Figure 7: Comparison of PPC results between log-normal (red) and exponential (green) distributions: outdegree distribution (*upper left*), indegree distribution (*upper right*), receiver size distribution (*lower left*), and PPplot for time increments (*lower right*). Blue lines denote the observed statistics (in the first three plots) or the diagonal line connecting $(0, 0)$ and $(1, 1)$ (in the last plot).

Appendix D: Convergence Diagnostics

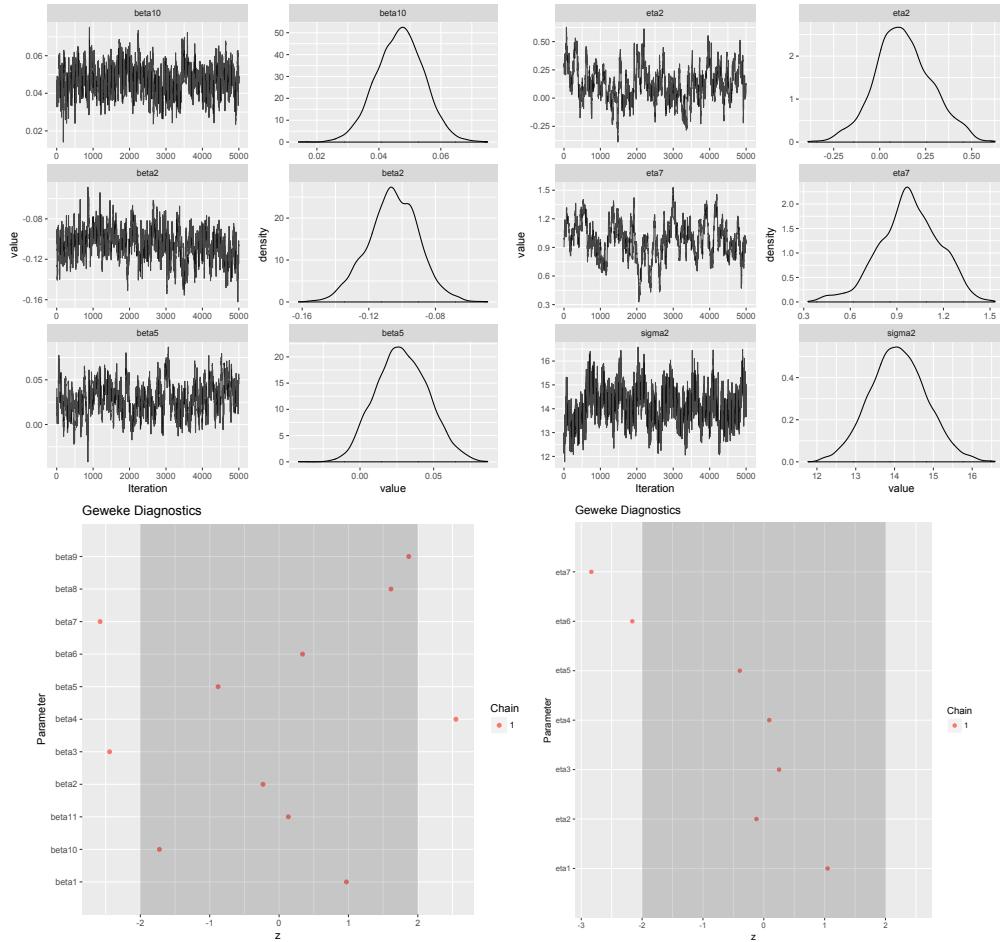


Figure 8: Convergence diagnostics from log-normal distribution: traceplots and density plots for β chains (upper left) and η chains (upper right), and geweke diagnostics for β chains (lower left) and η chains (lower right).

Supplementary Material

Title of the Supplement A (<http://www.some-url-address.org/dowload/0000.zip>). Add description for supplement material.

References

- Altman, M., Gill, J., and McDonald, M. P. (2004). *Numerical issues in statistical computing for the social scientist*, volume 508. John Wiley & Sons. [7](#)
- ben Aaron, J., Denny, M., Desmarais, B., and Wallach, H. (2017). “Transparency by Conformity: A Field Experiment Evaluating Openness in Local Governments.” *Public Administration Review*, 77(1): 68–77. [9](#)
- Blattner, M. (2009). “B-rank: A top N recommendation algorithm.” *arXiv preprint arXiv:0908.2741*. [2](#)
- Burgess, A., Jackson, T., and Edwards, J. (2004). “Email overload: Tolerance levels of employees within the workplace.” In *Innovations Through Information Technology: 2004 Information Resources Management Association International Conference, New Orleans, Louisiana, USA, May 23-26, 2004*, volume 1, 205. IGI Global. [1](#)
- Butts, C. T. (2008). “A RELATIONAL EVENT FRAMEWORK FOR SOCIAL ACTION.” *Sociological Methodology*, 38(1): 155–200. [1, 18](#)
- Dai, B., Ding, S., Wahba, G., et al. (2013). “Multivariate bernoulli distribution.” *Bernoulli*, 19(4): 1465–1483. [3](#)
- Desmarais, B. A. and Cranmer, S. J. (2017). “Statistical Inference in Political Networks Research.” In Victor, J. N., Montgomery, A. H., and Lubell, M. (eds.), *The Oxford Handbook of Political Networks*. Oxford University Press. [1](#)
- Fan, Y. and Shelton, C. R. (2009). “Learning continuous-time social network dynamics.” In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 161–168. AUAI Press. [2](#)
- Fellows, I. and Handcock, M. (2017). “Removing Phase Transitions from Gibbs Measures.” In *Artificial Intelligence and Statistics*, 289–297. [3](#)
- Geweke, J. (2004). “Getting it right: Joint distribution tests of posterior simulators.” *Journal of the American Statistical Association*, 99(467): 799–804. [5, 7](#)
- Geweke, J. et al. (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, volume 196. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN, USA. [16](#)
- Ghoshal, G., Zlatić, V., Caldarelli, G., and Newman, M. (2009). “Random hypergraphs and their applications.” *Physical Review E*, 79(6): 066118. [2](#)
- Hunter, D., Smyth, P., Vu, D. Q., and Asuncion, A. U. (2011). “Dynamic egocentric models for citation networks.” In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 857–864. [1](#)
- Hyndman, R. J. and Koehler, A. B. (2006). “Another look at measures of forecast accuracy.” *International journal of forecasting*, 22(4): 679–688. [14](#)
- Kanungo, S. and Jain, V. (2008). “Modeling email use: a case of email system transition.” *System Dynamics Review*, 24(3): 299–319. [1](#)

- Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S. (1999). "Multilevel hypergraph partitioning: applications in VLSI domain." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 7(1): 69–79. [2](#)
- Klamt, S., Haus, U.-U., and Theis, F. (2009). "Hypergraphs and cellular networks." *PLoS computational biology*, 5(5): e1000385. [2](#)
- McCullough, B. D. (2009). "The accuracy of econometric software." *Handbook of computational econometrics*, 55–79. [7](#)
- Neal, P. and Kypraios, T. (2015). "Exact Bayesian inference via data augmentation." *Statistics and Computing*, 25(2): 333–347. [6](#)
- Nelder, J. A. and Baker, R. J. (1972). *Generalized linear models*. Wiley Online Library. [4](#)
- Perry, P. O. and Wolfe, P. J. (2013). "Point process modelling for directed interaction networks." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(5): 821–849. [2](#), [11](#)
- Pew, R. C. (2016). "Social Media Fact Sheet." Accessed on 03/07/17. [1](#)
- Rao, P. (2000). "Applied survival analysis: regression modeling of time to event data." *Journal of the American Statistical Association*, 95(450): 681–681. [4](#)
- Rizopoulos, D. (2012). *Joint models for longitudinal and time-to-event data: With applications in R*. CRC Press. [4](#)
- Rubin, D. B. et al. (1984). "Bayesianly justifiable and relevant frequency calculations for the applied statistician." *The Annals of Statistics*, 12(4): 1151–1172. [15](#)
- Snijders, T., Steglich, C., and Schweinberger, M. (2007). *Modeling the coevolution of networks and behavior*. na. [1](#), [12](#)
- Snijders, T. A. (1996). "Stochastic actor-oriented models for network change." *Journal of mathematical sociology*, 21(1-2): 149–172. [1](#), [4](#), [12](#)
- Szóstek, A. M. (2011). "?Dealing with My Emails?: Latent user needs in email management." *Computers in Human Behavior*, 27(2): 723–729. [1](#)
- Tanner, M. A. and Wong, W. H. (1987). "The calculation of posterior distributions by data augmentation." *Journal of the American statistical Association*, 82(398): 528–540. [6](#)
- Vu, D. Q., Hunter, D., Smyth, P., and Asuncion, A. U. (2011). "Continuous-Time Regression Models for Longitudinal Networks." In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems 24*, 2492–2500. Curran Associates, Inc. [1](#)
- Zhang, Z.-K. and Liu, C. (2010). "A hypergraph model of social tagging networks." *Journal of Statistical Mechanics: Theory and Experiment*, 2010(10): P10005. [2](#)
- Zhang, Z.-K., Zhou, T., and Zhang, Y.-C. (2010). "Personalized recommendation via

- integrated diffusion on user–item–tag tripartite graphs.” *Physica A: Statistical Mechanics and its Applications*, 389(1): 179–186. [2](#)
- Zhao, T., Wang, Z., Cumberworth, A., Gsponer, J., de Freitas, N., Bouchard-Côté, A., et al. (2016). “Bayesian analysis of continuous time Markov chains with application to phylogenetic modelling.” *Bayesian Analysis*, 11(4): 1203–1237. [7](#)
- Zlatić, V., Ghoshal, G., and Caldarelli, G. (2009). “Hypergraph topological quantities for tagged social networks.” *Physical Review E*, 80(3): 036118. [2](#)