A Hyper-edge Event Model

Bomin Kim*
Department of Statistics, Pennsylvania State University,
Aaron Schein
College of Information and Computer Sciences, UMass Amherst,
Bruce Desmarais
Department of Political Science, Pennsylvania State University,
and
Hanna Wallach
Microsoft Research NYC

May 14, 2018

Abstract

We introduce the hyper-edge event model (HeEM)—a generative model for directed edges with multiple recipients or multiple senders. To define the model, we integrate a dynamic version of the exponential random graph model (ERGM) and generalized linear model (GLM) approach to jointly understand who communicates with whom, and when. We use the model to analyze emails sent between 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 recipients.

Keywords: 3 to 6 keywords, that do not appear in the title

*The authors gratefully acknowledge *please remember to list all relevant funding sources in the unblinded version*

1 Introduction

In recent decades, real-time digitized textual communication has developed into a ubiquitous form of social and professional interaction (Kanungo & 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 text exchanged in continuous time. A number of models that build upon topic modeling through Latent Dirichlet Allocation (Blei et al. 2003) to incorporate link data as well as textual content have been developed recently (McCallum et al. 2005, Lim et al. 2013, Krafft et al. 2012). These models are innovative in their extensions that incorporate network information. However, none of the models that are currently available in the literature integrate the rich random-graph structure offered by state of the art models for network structure—such as the exponential random graph model (ERGM) (Robins et al. 2007, Chatterjee et al. 2013, Hunter et al. 2008). The ERGM is the canonical model for modeling the structure of a static network. It is flexible enough to specify a generative model that accounts for nearly any pattern of edge formation such as reciprocity, clustering, popularity effects (Desmarais & Cranmer 2017).

Several network models have been developed that handle time-stamped events in which edge formation is governed by structural dynamics similar to those used in the ERGM (Butts 2008, Vu et al. 2011, Snijders 1996). Those models are very useful to understand which traits and behaviours are predictive of interactions, however, none of the models explicitly allows hyper-edges—a connection between two or more vertices generated simultaneously[1]—which is a common property of digitalized textual interactions such as emails and online messages. For instance, Perry & Wolfe (2013) treat multicast interactions—one type of directed hyper-edge which involve one sender and multiple receivers—via duplication (i.e., obtain pairwise interactions from the original multicast), but their model parameter estimation relies on likelihood approximation which leads to bias. Similarly, Fan & Shelton (2009) treats multicast emails as multiple single-recipient edges and randomly jitter the sent times, to avoid the violation of continous-time model assumption. Concentrating on better handling of hyper-edges, we develop the hyper-edge event model (HeEM)

[1] A hyper-edge connecting just two nodes is simply a usual dyadic edge.

which simultaneously models the two components that govern time-stamped event formation: 1) the recipient selection processs that allows multiple senders or receivers, and 2) the time-to-next interactions with flexible distributional choices.

In what follows, we introduce the HeEM by describing how we assume the generatve process of a time-stamped event data (Section 2), and deriving the sampling equations for Bayesian inference (Section 3). Then, we apply the model to the Montgomery county government email data and perform two model validation tasks (Section 4). Finally, the paper finishes in Section 5 with conclusion and discussion.

2 A Hyper-Edge Event Model (HeEM)

Data generated under the model consists of D unique edges. A single edge, indexed by $d \in [D]$, is represented by the three components: the sender $a_d \in [A]$, an indicator vector of recipients $\boldsymbol{r}_d = \{u_{dr}\}_{r=1}^{A}$, and the 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 hyper-edges—edges with (1) a single sender and multiple receivers, and (2) multiple senders and a single receiver—here we only present the generative process for those involving a single sender and multiple receivers (i.e., multicast). For the latter case of hyper-edges, we treat a_d to be an indicator vector of senders $\boldsymbol{a}_d = \{u_{dr}\}_{a=1}^{A}$ and r_d to be the single recipient.

2.1 Edge Generating Process

For every possible author–recipient pair $(a, r)_{a \neq r}$, we define the "recipient intensity", which is the likelihood of edge d being sent from a to r:

$$\lambda_{adr} = \boldsymbol{b}^{\top} \boldsymbol{x}_{adr}, \tag{1}$$

where $\boldsymbol{b}$ is P–dimensional vector of coefficients and $\boldsymbol{x}_{adr}$ is a set of network 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 recipients. We place a Normal prior $\boldsymbol{b} \sim N(\boldsymbol{\mu}_b, \Sigma_b)$.

Next, we hypothesize "If a were the sender of edge d, who would be the recipient(s)?" To do this, we draw each sender's set of recipients from the multivariate Bernoulli (MB) distribution (Dai et al. 2013)—a model to estimate the structure of graphs with binary nodes—with probability of 1 being logit(λ_{adr}). In order to avoid the model degeneracy from having an empty recipient set, we define a probability measure "MB_{Gibbs}" motivated by the non-empty Gibbs measure (Fellows & Handcock 2017) which excludes the all-zero vector from the support of MB distirbution. As a result, we 1) allow multiple recipients or "multicast", 2) prevent from obtaining zero recipient, and 3) ensure tractable normalizing constant. To be specific, we draw a binary vector $\boldsymbol{u}_{ad} = (u_{ad1}, \ldots, u_{adA})$

$$\boldsymbol{u}_{ad} \sim \text{MB}_{Gibbs}(\boldsymbol{\lambda}_{ad}), \tag{2}$$

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

$$p(\boldsymbol{u}_{ad}|\delta, \boldsymbol{\lambda}_{ad}) = \frac{\exp\left\{\log\big(\text{I}(\|\boldsymbol{u}_{ad}\|_1 > 0)\big) + \sum_{r \neq a} \lambda_{adr} u_{adr}\right\}}{Z(\boldsymbol{\lambda}_{ad})}, \tag{3}$$

where $Z(\boldsymbol{\lambda}_{ad}) = \prod_{r \neq a} \big(\exp(\lambda_{adr}) + 1\big) - 1$ is the normalizing constant and $\|\cdot\|_1$ is the l_1–norm. Again, this is equivalent to assume independent Bernoulli trial on each u_{adr} with probability of 1 being logit(λ_{adr}), excluding the case when all $u_{adr} = 0$. We provide the derivation of the normalizing constant as a tractable form in the supplementary material.

2.2 Time Generating Process

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

$$\mu_{ad} = g^{-1}(\boldsymbol{\eta}^\top \boldsymbol{y}_{ad}), \tag{4}$$

where $\boldsymbol{\eta}$ is Q–dimensional vector of coefficients with a Normal prior $\boldsymbol{\eta} \sim N(\boldsymbol{\mu}_\eta, \Sigma_\eta)$, $\boldsymbol{y}_{ad}$ is a set of time-related covariates, e.g., any feature 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 the time-increment or "time to next interaction" (i.e., $\tau_d = t_d - t_{d-1}$)

is drawn from a specific distribution in the exponential family. We follow the generalized linear model (GLM) framework (Nelder & Baker 1972):

$$\begin{aligned} E(\tau_{ad}) &= \mu_{ad}, \\ V(\tau_{ad}) &= V(\mu_{ad}), \end{aligned} \tag{5}$$

where τ_{ad} here is a positive real number. Possible choices of distribution include Exponential, Weibull, Gamma, and lognormal[2] 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 parameter (e.g., σ_τ^2) to account for the variance. We use $f_\tau(\cdot; \mu, \sigma_\tau^2)$ and $F_\tau(\cdot; \mu, \sigma_\tau^2)$ to denote the probability density function (p.d.f) and cumulative density function (c.d.f), respectively, with mean μ and variance σ^2.

2.3 Observed Data

Finally, we choose the sender, recipients, and timestamp—which will be observed—by selecting the sender–recipient-set pair with the smallest time-increment (Snijders 1996):

$$\begin{aligned} a_d &= \text{argmin}_a(\tau_{ad}), \\ \boldsymbol{r}_d &= \boldsymbol{u}_{a_d d}, \\ t_d &= t_{d-1} + \tau_{a_d d}. \end{aligned} \tag{6}$$

Therefore, it is a sender-driven process in that the recipients and timestamp of an edge is determined by the sender's urgency to send the edge to chosen recipients. 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 τ_d), we observe all of the tied events without assigning the orders of tied events. Algorithm 1 summarizes the generative process in Section 2.

3 Posterior Inference

Our inference goal is to invert the generative process to obtain the posterior distribution over the unknown parameters, conditioned on the observed data and hyperparamters

[2]lognormal distribution is not exponential family but can be used via modeling of $\log(\tau_d)$.

Algorithm 1 Generating Process

for *d=1 to D* **do**
  for *a=1 to A* **do**
    for *r=1 to A (r ≠ a)* **do**
      set $\lambda_{adr} = \boldsymbol{b}^\top \boldsymbol{x}_{adr}$
    end
    draw $\boldsymbol{u}_{ad} \sim \text{Gibbs}(\boldsymbol{\lambda}_{ad})$
    set $\mu_{ad} = g^{-1}(\boldsymbol{\eta}^\top \boldsymbol{y}_{ad})$
    draw $\tau_{ad} \sim f_\tau(\mu_{ad}, \sigma_\tau^2)$
  end
  For no tied events,
  set $a_d = \text{argmin}_a(\tau_{ad})$
  set $\boldsymbol{r}_d = \boldsymbol{u}_{a_d d}$
  set $t_d = t_{d-1} + \min_a \tau_{ad}$

  For n **tied events,** (i.e., n senders generated same minimum time increments)
  set $a_d, \ldots, a_{d+n} = \text{argmin}_a(\tau_{ad})$
  set $\boldsymbol{r}_d = \boldsymbol{u}_{a_d d}, \ldots, \boldsymbol{r}_{d+n} =, \boldsymbol{u}_{a_{d+n} d}$
  set $t_d, \ldots, t_{d+n} = t_{d-1} + \min_a \tau_{ad}$
  jump to $d = d + n + 1$
end

$(\boldsymbol{\mu}_b, \Sigma_b, \boldsymbol{\mu}_\eta, \Sigma_\eta)$. We draw the samples using Markov chain Monte Carlo (MCMC) methods, repeatedly resampling the value of each parameter from its conditional posterior given the observed data, hyperparamters, and the current values of the other parameters. We express each parameters conditional posterior in a closed form using the data augmentation schemes in $\boldsymbol{u}$ (Tanner & Wong 1987). In this section, we provide each latent variable's conditional posterior.

First, since u_{adr} is a binary random variable, new values may be sampled directly using

$$\begin{aligned} P(u_{adr} = 1 | \boldsymbol{u}_{ad\setminus r}, \boldsymbol{b}, \boldsymbol{x}) &\propto \exp(\lambda_{adr}); \\ P(u_{adr} = 0 | \boldsymbol{u}_{ad\setminus r}, \boldsymbol{b}, \boldsymbol{x}) &\propto \text{I}(\|\boldsymbol{u}_{ad\setminus r}\|_1 > 0), \end{aligned} \tag{7}$$

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

New values for continuous variables $\boldsymbol{b}$, and $\boldsymbol{\eta}$ and σ_τ^2 (if applicable) cannot be sampled directly from their conditional posteriors, but may instead be obtained using the Metropolis–Hastings algorithm. With uninformative priors (i.e., $N(0, \infty)$), the conditional

posterior over $\boldsymbol{b}$ is

$$P(\boldsymbol{b}|\boldsymbol{u},\boldsymbol{x}) \propto \prod_{d=1}^{D}\prod_{a=1}^{A} \frac{\exp\left\{\log\big(\mathrm{I}(\|\boldsymbol{u}_{ad}\|_1 > 0)\big) + \sum_{r\neq a}\lambda_{adr}u_{adr}\right\}}{Z(\boldsymbol{\lambda}_{ad})}, \tag{8}$$

where the two variables share the conditional posterior and thus can be jointly sampled. Likewise, assuming uninformative priors on $\boldsymbol{\eta}$ (i.e., $N(0,\infty)$) and σ_τ^2 (i.e., half-Cauchy(∞)), the conditional posterior for no-tied event case is

$$P(\boldsymbol{\eta},\sigma_\tau^2|\boldsymbol{u},\boldsymbol{y}) \propto \prod_{d=1}^{D}\Big(f_\tau(\tau_d;\mu_{a_d d},\sigma_\tau^2) \times \prod_{a\neq a_d}\big(1-F_\tau(\tau_d;\mu_{ad},\sigma_\tau^2)\big)\Big), \tag{9}$$

where $f_\tau(\tau_d;\mu_{a_d d},\sigma_\tau^2)$ is the probability that the d^{th} observed time-increment comes from the specified distribution $f_\tau(\cdot)$ with the observed sender's mean parameter, and $\prod_{a\neq a_d}\big(1-F_\tau(\tau_d;\mu_{ad},\sigma_\tau^2)\big)$ is the probability that the latent (or unobserved) senders for event d all generated time-increments greater than τ_d. Moreover, under the existence of tied-event, the conditional posterior of $\boldsymbol{\eta}$ and σ_τ^2 should be written as

$$P(\boldsymbol{\eta},\sigma_\tau^2|\boldsymbol{u},\boldsymbol{y}) \propto \prod_{m=1}^{M}\Big(\prod_{d:t_d=t_m^*} f_\tau(t_m^*-t_{m-1}^*;\mu_{a_d d},\sigma_\tau^2)\times \prod_{a\notin\{a_d\}_{d:t_d=t_m^*}}\big(1-F_\tau(t_m^*-t_{m-1}^*;\mu_{ad},\sigma_\tau^2)\big)\Big), \tag{10}$$

where $t_1^*,\ldots,t_M^*$ are the unique timepoints across D events ($M \leq D$). If $M = D$ (i.e., no tied events), Equation (10) reduces to Equation (9). Algorithm 2 provides the pseudocode.

Algorithm 2 MCMC Algorithm

set initial values of $\boldsymbol{b}, \boldsymbol{\eta}$, (and σ_τ^2)
for *o=1 to outer* **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 (7)
      end
    end
  end
  for *n=1 to inner1* **do**
    update $\boldsymbol{b}$ using Metropolis-Hastings — Equation (8)
  end
  for *n=1 to inner2* **do**
    update $\boldsymbol{\eta}$ using Metropolis-Hastings — Equation (10) (if needed) update σ_τ^2 using Metropolis-Hastings — Equation (10)
  end
end
Summarize the results with:
last chain of $\boldsymbol{b}$, and last chain of $\boldsymbol{\eta}$ (and σ_τ^2)

4 Application to Montgomary County Emails

We now present a case study applying our method to Montgomery county government email data. For this case study, we formulate the network statistics $\boldsymbol{x}$ and timestamp statistics $\boldsymbol{y}$ and ground them in illustrative examples. We then report a suite of experiments that test our methods ability to form the posterior distribution over latent variables.

4.1 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 a lot of multicast edges (16.76%), and 2) limit the scope of this initial application. The Montgomery County email network contains 680 emails, sent and received by 18 department managers over a period of 3 months (March–May) in 2012.

4.2 Covariates

In the example of email networks, we form the covariate vector $\boldsymbol{x}_{adr}$ using time-varying network statistics based on the time interval prior to and including t_{d-1}. Along with intercept term, we compute the network statistics (Perry & Wolfe 2013), where the time interval tracks 7 days prior to the last email was sent $l_d = (t_{d-1} - 7\text{days}, t_{d-1}]$. To be specific, we use

1. intercept $x_{adr1} = 1$;
2. outdegree $x_{adr2} = \sum_{d':t_{d'} \in l_d} I(a_{d'} = a)$;
3. indegree $x_{adr3} = \sum_{d':t_{d'} \in l_d} I(u_{d'r} = 1)$;
4. send $x_{adr4} = \sum_{d':t_{d'} \in l_d} I(a_{d'} = a) I(u_{d'r} = 1)$;
5. receive $x_{adr5} = \text{send}(r, a)$;
6. 2-send $x_{adr6} = \sum_{h \neq a,r} \text{send}(a, h)\text{send}(h, r)$;
7. 2-receive $x_{adr7} = \sum_{h \neq a,r} \text{send}(h, a)\text{send}(r, h)$;
8. sibling $x_{adr8} = \sum_{h \neq a,r} \text{send}(h, a)\text{send}(h, r)$;
9. cosibling $x_{adr9} = \sum_{h \neq a,r} \text{send}(a, h)\text{send}(r, h)$;
10. Nreceive $x_{adr10} = \sum_{d':t_{d'} \in l_d} \sum_{r=1}^{A} I(a_{d'} = a) I(u_{d'r} = 1)$;
11. outdegree$*$Nreceive $x_{adr11} = x_{adr2} \times x_{adr10}$;

where $I(\cdot)$ is an indicator function, and the last covariate is the interaction term between outdegree and Nreceive. As with the nodal (2, 3, 10, 11) and dyadic (4, 5) effects, the triadic (6–9) effects are designed so that their coefficient have a straightforward interpretation, which are illustrated in Figure 1.

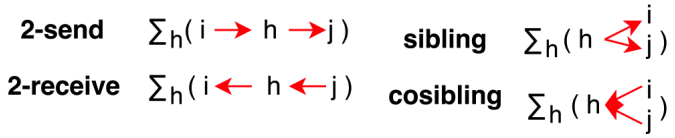

Figure 1: Visualization of triadic statistics.

For time-related covariates $\boldsymbol{y}_{ad}$ in Section 2.2, we form the covariate vector $\boldsymbol{y}_{ad}$ using the seven statistics which may possibly effect "time to send the next document". Note that these statistics could depend on a only (nodal effect), d (or t_d) only (temporal effect), or both a and d (nodal & temporal effect). Specifically, the statistics are defined as

1. intercept $y_{ad1} = 1$;
2. outdegree $y_{ad2} = \sum_{d':t_{d'} \in l_d} I(a_{d'} = a)$;
3. indegree $y_{ad3} = \sum_{d':t_{d'} \in l_d} I(u_{d'a} = 1)$;
4. gender $y_{ad4} = I(a = \text{female})$;
5. manager $y_{ad5} = I(a = \text{County Manager})$;
6. weekend $y_{ad6} = I(t_{d-1} = \text{weekend})$;
7. am/pm $y_{ad7} = I(t_{d-1} = \text{pm})$.

4.3 Parameter Estimation and Interpretation

Based on some preliminary experiments, we choose lognormal distribution to model time-to-next-email in Montgomery county email data, thus we estimate the variance parameter σ_τ^2 with the prior specified as inverse-Gamma$(2, 1)$. In this section, we present the results based on 55,000 MCMC outer iterations with a burn-in of 15,000, where we thin by keeping every 40th sample. While the inner iterations for $\boldsymbol{\eta}$ and σ_τ^2 are fixed as 1, we use 10 inner iterations for $\boldsymbol{b}$ to adjust for its slower convergence rate.

Figure 2 shows the boxplots summarizing posterior samples of $\boldsymbol{b}$. Since we use logit probability of λ_{adr} for edge generating process (Section 2.1), we have

$$\text{logit}(\lambda_{adr}) = \log\left(\frac{\lambda_{adr}}{1-\lambda_{adr}}\right) = b_1 + b_2 x_{adr2} \ldots + b_{11} x_{adr11},$$

and can interpret the parameter estimates in terms of odds ratio $\frac{\lambda_{adr}}{1-\lambda_{adr}} = \exp(b_1 + b_2 x_{adr2} \ldots + b_{11} x_{adr11})$. Firstly, we can see that the effect of send (i.e., "number of times the actor a sent emails to the actor r over the last week") is positive, implying that if the actor a sent n number of emails to r last week, then the actor 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 statistics indegree and Nrecieve. For example, as the actor r received n emails and the actor a sent emails to n rececievers over the last week, the actor a is $\exp(0.086 \times n) \approx (1.091)^n$ times and $\exp(0.047 \times n) \approx (1.048)^n$ times, respectively, more likely to send an email to r. On the other hand, outdegree statistic has negative effect—i.e., if the actor a sent n number of emails to anyone last week, then the actor a is approximately $\exp(-0.109 \times n) \approx (0.897)^n$ times likely to send an email to r—possibly due to its multicollinearity with the statistics send and Nreceive. This can be interpreted as "if the actor a sent large number of emails over the last week but 1) not to r (excluding the effect of send) or 2) no broadcast or multicast emails (excluding the effect of Nreceive)" then the actor a is less likely to send an email to r. Rest of statistics are not statistically significant because their 95% credible interval do not include 0. Surprisingly, none of the triadic effects seem to have significant effect on the edge generating process.

Figure 3 shows the boxplots summarizing posterior samples of the time-related coefficients $\boldsymbol{\eta}$. For this specific datset, we assume lognormal distribution (Section 2.2) with the

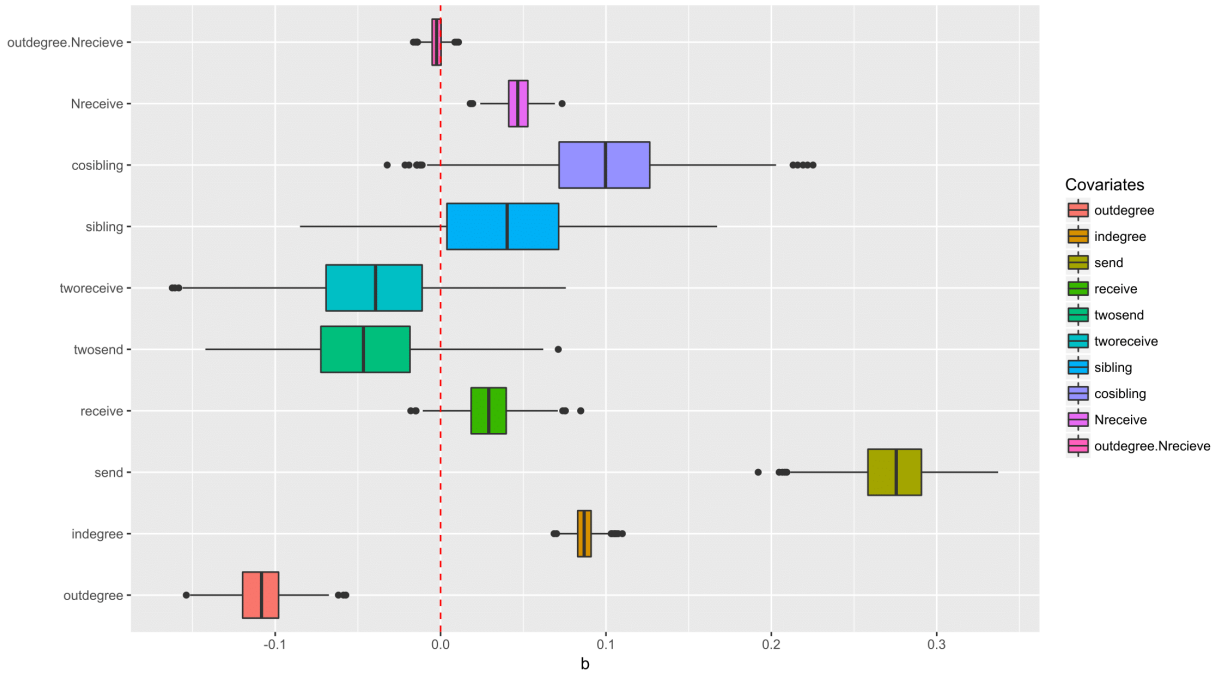

Figure 2: Posterior distribution of $\boldsymbol{b}$ estimates.

unit of time being an hour, so that the interpretation of $\hat{\boldsymbol{\eta}}$ should be based on

$$\log(\tau_{ad}) \sim N(\mu_{ad}, \sigma_\tau^2), \text{ with}$$

$$\mu_{ad} = \eta_1 + \eta_2 y_{ad2} \ldots + \eta_7 y_{ad7}.$$

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.

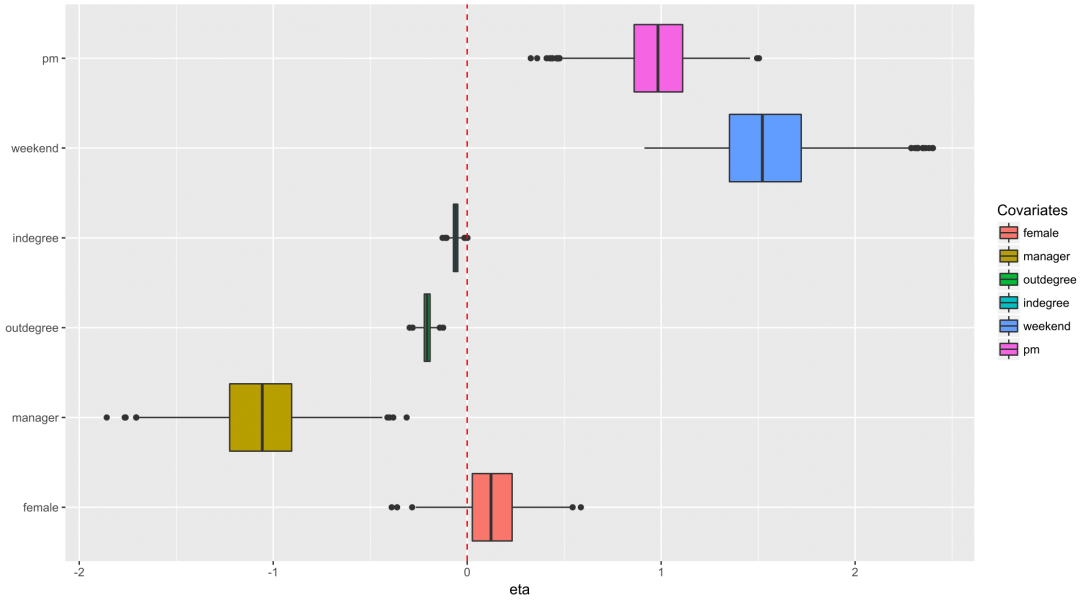

Figure 3: Posterior distribution of $\boldsymbol{\eta}$ estimates.

4.4 Posterior Predictive Checks

We perform posterior predictive checks (Rubin et al. 1984) to evaluate the appropriateness of our model specification for the Montgomery County email data. We formally generated entirely new data, by simulating time-stamped events $\{(a_d, \boldsymbol{r}_d, t_d)\}_{d=1}^{D}$ from the genenerative process in Section 2, conditional upon a set of inferred parameter values from the inference in Section 4.3. For the test of goodness-of-fit in terms of network dynamics, we use multiple network statistics that summarize meaningful aspects of the Montgomery County email data: outdegree distribution, indegree distribution, recipient size distribution, and time-increments probability-probability (PP) plot. We then generated 500 synthetic networks from the posterior predictive distribution, according to Algorithm 3.

Figure 4 illustrates the results of poterior predictive checks. Upper two plots show node-specific degree distributions, where the left one is the boxplots of 500 posterior predictive outdegree statistic and the right plot is the same one for indegree statistic. For both plots, the x-axis represents the node id's from 1 to 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, without using node-specific indicators. Outdegree distribution of some low-activity nodes are not precisely recovered, however, indegree distribution looks much better. Since we use more information in the recipient selection process (i.e., network effects) while we only rely on minimum time-increments when choosing the ob-

Algorithm 3 Generate new data for PPC

Input: number of new data to generate R, covariates $\boldsymbol{x}$ and $\boldsymbol{y}$
estimated latent variables $(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{\eta}, \sigma_\tau^2)$

for $r = 1$ **to** R **do**
  for $d = 1$ **to** D **do**
    Draw $(a_d, \boldsymbol{r}_d, t_d)$ following Section 2
  end for
  Store every r^{th} new data $\{(a_d, \boldsymbol{r}_d, t_d)\}_{d=1}^{D}$
end for

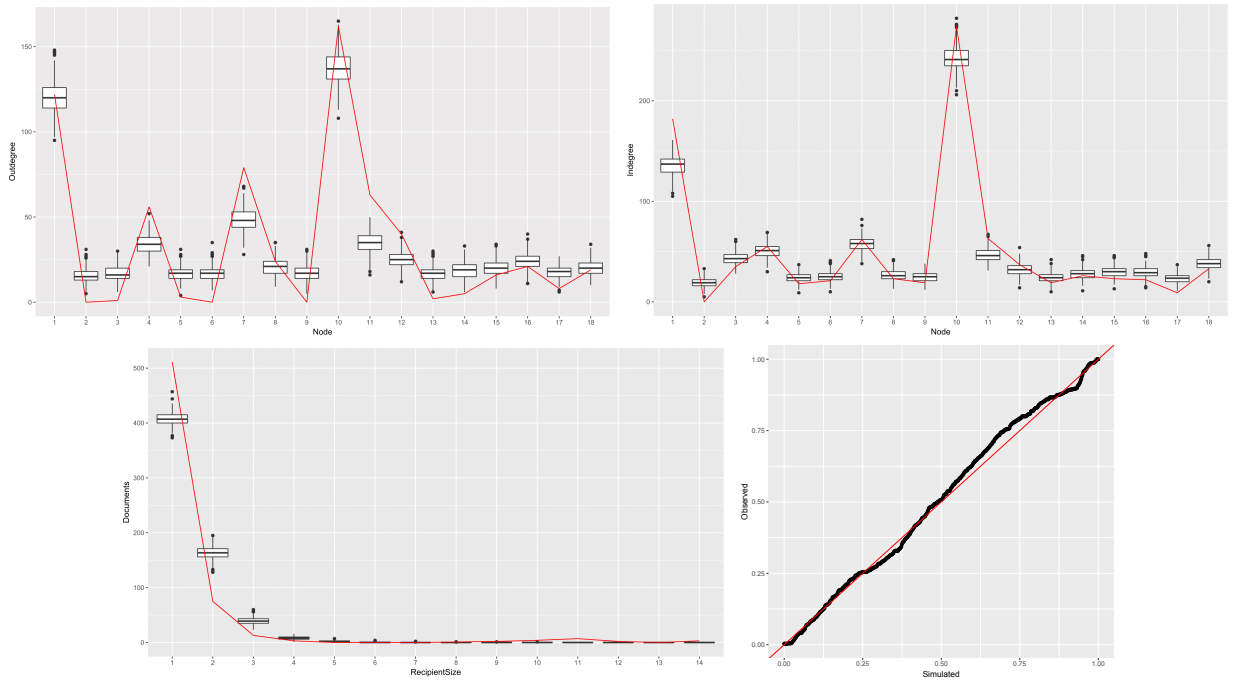

Figure 4: PPC results from lognormal distribution: outdegree distribution (*upper left*), indegree distribution (*upper right*), recipient size distribution (*lower left*), and time-increments probability-probability (PP) plot (*lower right*). Red lines in the first three plot depict the observed statistics, and the red line in the last plot is the diagonal line connecting $(0, 0)$ and $(1, 1)$.

served sender, these results are expected. Lower left plot is the distribution of recipient sizes, where the x-axis spans over the size of recipients 1 to 14 (which is the maximum size of observed recipients) and the y-axis denotes the number of documents with x-number of recipients. The result shows that our model is underestimating single-receiver documents while overestimating documents with two, three, and four recipients. One explanation behind what we observe is that the model is trying to recover so-called "broadcast" emails, which are the documents with ≥ 10 number of recipients, so that the intercept estimate b_1 is slightly moved toward right. To improve the goodness of fit for recipient 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 is a graphical measure commonly used for assessing how closely two data sets agree, by plotting the two cumulative distribution functions against each other. The resulting goodness of fit of 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.

4.5 Prediction Experiment

The HeEM has many component parts that need to be specified by the user (i.e., the selection of the event timing features $\boldsymbol{y}$, the recipient selection features $\boldsymbol{x}$, and the event time distribution f). Many of these components will be specified based on user expertise (e.g., regarding which features would drive recipient 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 HeEM specification using out-of-sample prediction. We illustrate this approach by comparing alternative distributional families for the event timing component of the model.

We evaluate the model's ability to predict edges and timestamps from the 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 components—sender, recipients, 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, parameter 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 4 outlines this procedure in detail.

Here, we specifically compare the predictive performance from two distributions—lognormal and exponential. We particularly choose exponential distribution as an alternative to what we used earlier (i.e., lognormal) since exponential is the most commonly

Algorithm 4 Out-of-Sample Predictions

Input: data $\{(a_d, \boldsymbol{r}_d, t_d)\}_{d=1}^{D}$, number of new data to generate R, hyperparameters

Test splits:
draw test authors with $p = 0.1$ (out of D authors)
draw test recipients with $p = 0.1$ (out of $D \times (A-1)$ recipient indicators $\{\{\boldsymbol{r}_{dr}\}_{r \in [A]_{\setminus a_d}}\}_{d=1}^{D}$)
draw test timestamps with $p = 0.1$ (out of D timestamps)
set the "test" data as "missing" (NA)

Imputation and inference:
initialize the parameters $(\boldsymbol{b}, \boldsymbol{\eta}, \boldsymbol{u}, \sigma_\tau^2)$
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 π_a using $P(a_d = a|\cdot) = f_\tau(\tau_d; \mu_{a_d d}, \sigma_\tau^2) \times \prod_{a \neq a_d} \left(1 - F_\tau(\tau_d; \mu_{ad}, \sigma_\tau^2)\right)$
      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} using $P(r_{dr} = 1|\cdot)$ and $P(r_{dr} = 0|\cdot)$—Equation (7)
      end if
    end for
    if $t_d = \text{NA}$ **then**
      draw $\boldsymbol{\tau}_d^{new}$ from $f_\tau(\tau_d^{new}; \mu_{a_d d}, \sigma_\tau^2) \times \prod_{a \neq a_d} \left(1 - F_\tau(\tau_d; \mu_{ad}, \sigma_\tau^2)\right)$ via importance sampling (e.g., proposal distribution $g \sim \text{halfcauchy}(5)$)
    end if
    run inference and update $(\boldsymbol{b}, \boldsymbol{\eta}, \boldsymbol{u}, \sigma_\tau^2)$ given the imputed and observed data
  end for
  store the estimates for "test" data
end for

specified distribution for time-to-event data which is also used in the stochastic actor oriented models (Snijders 1996) as well as their extensions (Snijders et al. 2007). We run the experiment and measure the predictive performance of two separate time distributions using 500 predicted samples, which are depicted in Figure 5. First, we compare the correct sender probability for missing documents $\{d : a_d = \text{NA}\}$, which corresponds to π_{a_d} in Algorithm 4. The plot on the left shows that lognormal distribution reaches higher correct sender probability than exponential distribution, although their centers (e.g., medians) are similar. Secondly, we compute similar measure for recipients—correct recipient probability

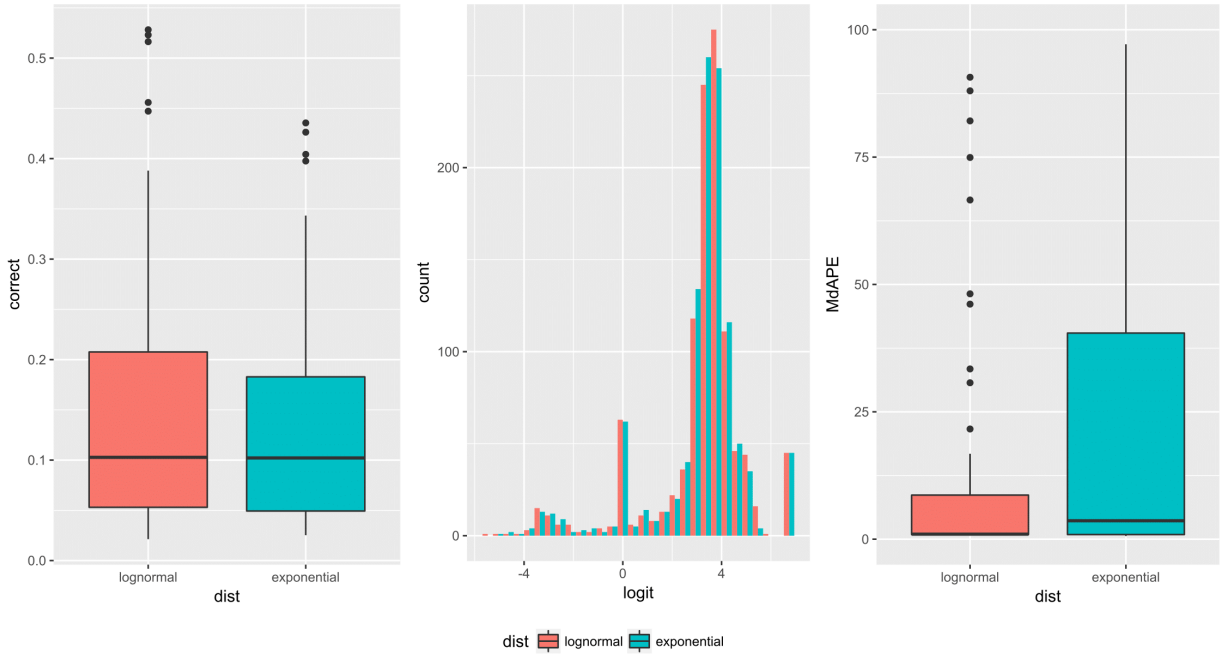

Figure 5: Comparison of predictive performance between lognormal and exponential distributions: boxplots for correct sender probability (*left*), distribution of correct recipient probability in logit (*middle*), and boxplots for median absolute percentage error (*right*).

$P(r_{dr} = r_{dr}^{obs})$ in Algorithm 4 where r_{dr}^{obs} is either 1 (received) or 0 (not received)—and visually compare the two distributions in logit scale (i.e., $\log(\frac{P(r_{dr}=r_{dr}^{obs})}{1-P(r_{dr}=r_{dr}^{obs})})$). Since the edge generating process (Section 2.1) is not directly affected by the choice of timestamp distribution, there is no significant difference between lognormal and exponential in their performance of predicting unknown recipients. Finally, prediction errors for missing timestamps are measured using the median absolute percentage error (MdAPE) (Hyndman & Koehler 2006), where we take the median of absolute percentage errors p_{di} for $i = 1, \ldots, 500$ for dth document, calculated as follows:

$$p_{di} = \left| \frac{\tau_d^{obs} - \tau_{di}^{pred}}{\tau_d^{obs}} \right|,$$

and we present boxplots for the MdAPE estimates on the right plot. As expected, we have huge benefit in the performance of timestamp prediction when we choose lognormal distribution over exponential. This difference can be explained by our variance estimate $\hat{\sigma}_\tau^2 = 14.093$ in Section 4.3, because exponential distribution omits the extra information on

time-increments contained in the large variance estimate. All our findings in this section is further revealed in the PPC results from exponential distribution, which are attached in Appendix B. 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 information of the emails?" and 2) offer one standard way to determine the time-increment distribution in Section 2.2.

5 Conclusion

References

ben Aaron, J., Denny, M., Desmarais, B. & Wallach, H. (2017), 'Transparency by conformity: A field experiment evaluating openness in local governments', *Public Administration Review* **77**(1), 68–77.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *J. Mach. Learn. Res.* **3**, 993–1022.

Burgess, A., Jackson, T. & 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', Vol. 1, IGI Global, p. 205.

Butts, C. T. (2008), 'A relational event framework for social action', *Sociological Methodology* **38**(1), 155–200.

Chatterjee, S., Diaconis, P. et al. (2013), 'Estimating and understanding exponential random graph models', *The Annals of Statistics* **41**(5), 2428–2461.

Dai, B., Ding, S., Wahba, G. et al. (2013), 'Multivariate bernoulli distribution', *Bernoulli* **19**(4), 1465–1483.

Desmarais, B. A. & Cranmer, S. J. (2017), Statistical inference in political networks research, *in* J. N. Victor, A. H. Montgomery & M. Lubell, eds, 'The Oxford Handbook of Political Networks', Oxford University Press.

Fan, Y. & Shelton, C. R. (2009), Learning continuous-time social network dynamics, *in* 'Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence', AUAI Press, pp. 161–168.

Fellows, I. & Handcock, M. (2017), Removing phase transitions from gibbs measures, *in* 'Artificial Intelligence and Statistics', pp. 289–297.

Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M. & Morris, M. (2008), 'ergm:

A package to fit, simulate and diagnose exponential-family models for networks', *Journal of statistical software* **24**(3), nihpa54860.

Hyndman, R. J. & Koehler, A. B. (2006), 'Another look at measures of forecast accuracy', *International journal of forecasting* **22**(4), 679–688.

Kanungo, S. & Jain, V. (2008), 'Modeling email use: a case of email system transition', *System Dynamics Review* **24**(3), 299–319.

Krafft, P., Moore, J., Desmarais, B. & Wallach, H. M. (2012), Topic-partitioned multinetwork embeddings, *in* F. Pereira, C. Burges, L. Bottou & K. Weinberger, eds, 'Advances in Neural Information Processing Systems 25', Curran Associates, Inc., pp. 2807–2815.

Lim, K. W., Chen, C. & Buntine, W. (2013), Twitter-network topic model: A full bayesian treatment for social network and text modeling, *in* 'NIPS2013 Topic Model workshop', pp. 1–5.

McCallum, A., Corrada-Emmanuel, A. & Wang, X. (2005), The author-recipient-topic model for topic and role discovery in social networks, with application to enron and academic email, *in* 'Workshop on Link Analysis, Counterterrorism and Security', p. 33.

Nelder, J. A. & Baker, R. J. (1972), *Generalized linear models*, Wiley Online Library.

Perry, P. O. & 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.

Pew, R. C. (2016), 'Social media fact sheet', *Accessed on 03/07/17* .

Rao, P. (2000), 'Applied survival analysis: regression modeling of time to event data', *Journal of the American Statistical Association* **95**(450), 681–681.

Rizopoulos, D. (2012), *Joint models for longitudinal and time-to-event data: With applications in R*, CRC Press.

Robins, G., Pattison, P., Kalish, Y. & Lusher, D. (2007), 'An introduction to exponential random graph (p*) models for social networks', *Social networks* **29**(2), 173–191.

Rubin, D. B. et al. (1984), 'Bayesianly justifiable and relevant frequency calculations for the applied statistician', *The Annals of Statistics* **12**(4), 1151–1172.

Snijders, T. A. (1996), 'Stochastic actor-oriented models for network change', *Journal of mathematical sociology* **21**(1-2), 149–172.

Snijders, T., Steglich, C. & Schweinberger, M. (2007), *Modeling the coevolution of networks and behavior*, na.

Szóstek, A. M. (2011), '?dealing with my emails?: Latent user needs in email management.', *Computers in Human Behavior* **27**(2), 723–729.

Tanner, M. A. & Wong, W. H. (1987), 'The calculation of posterior distributions by data augmentation', *Journal of the American statistical Association* **82**(398), 528–540.

Vu, D. Q., Hunter, D., Smyth, P. & Asuncion, A. U. (2011), Continuous-time regression models for longitudinal networks, *in* J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira & K. Weinberger, eds, 'Advances in Neural Information Processing Systems 24', Curran Associates, Inc., pp. 2492–2500.

Appendices

A Normalizing Constant of MB_{Gibbs}

Our probability measure "MB_{Gibbs}"—the multivariate Bernoulli distribution with non-empty Gibbs measure (Fellows & Handcock 2017)—defines the probability of author a selecting the binary recipient vector $\boldsymbol{u}_{ad}$ as

$$P(\boldsymbol{u}_{ad}|\boldsymbol{\lambda}_{ad}) = \frac{\exp\left\{\log\big(\mathrm{I}(\|\boldsymbol{u}_{ad}\|_1 > 0)\big) + \sum_{r\neq a}\lambda_{adr}u_{adr}\right\}}{Z(\boldsymbol{\lambda}_{ad})}.$$

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

We recognize that if $\boldsymbol{u}_{ad}$ were drawn via independent Bernoulli distributions in which $P(u_{adr} = 1|\delta, \boldsymbol{\lambda}_{ad})$ was given by logit(λ_{adr}), then

$$P(\boldsymbol{u}_{ad}|\boldsymbol{\lambda}_{ad}) \propto \exp\Big\{\sum_{r\neq a} \lambda_{adr} u_{adr}\Big\}.$$

This is straightforward to verify by looking at

$$P(u_{adr} = 1|\boldsymbol{u}_{ad[-r]}, \boldsymbol{\lambda}_{ad}) = \frac{\exp(\lambda_{adr})}{\exp(\lambda_{adr}) + 1}.$$

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

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

Now, since

$$\begin{aligned}
&\exp\Big\{\log\Big(\mathrm{I}(\|\boldsymbol{u}_{ad}\|_1 > 0)\Big) + \sum_{r\neq a} \lambda_{adr} u_{adr}\Big\} \\
&= \exp\Big\{\sum_{r\neq a} \lambda_{adr} u_{adr}\Big\},
\end{aligned}$$

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

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

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

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

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

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

B PPC Results from Exponential Distribution

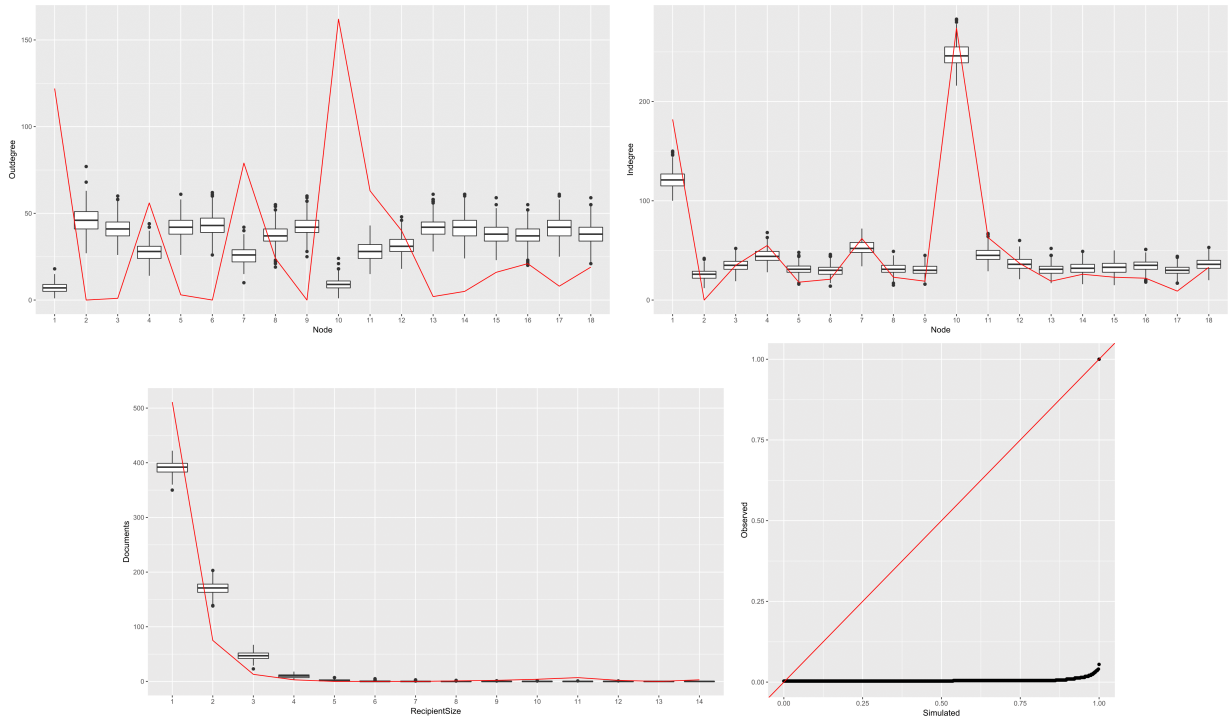

Figure 6: PPC results from exponential distribution: outdegree distribution (*upper left*), indegree distribution (*upper right*), recipient size distribution (*lower left*), and time-increments probability-probability (PP) plot (*lower right*). Red lines in the first three plot depict the observed statistics, and the red line in the last plot is the diagonal line connecting $(0, 0)$ and $(1, 1)$.