# Supplementary Material
# *Twitter-Network Topic Model: A Full Bayesian Treatment for Social Network and Text Modeling*

**Kar Wai Lim**
ANU, NICTA
Canberra, Australia

**Changyou Chen**
ANU, NICTA
Canberra, Australia

**Wray Buntine**
NICTA, ANU
Canberra, Australia

This supplementary material is to be used in conjunction with the paper *"Twitter-Network Topic Model: A Full Bayesian Treatment for Social Network and Text Modeling"*.

## Appendix A Generative Model of the Twitter-Network (TN) Topic Model

The TN topic model makes use of the accompanying *hashtags*, *authors*, and *followers network* to model tweets better. The TN topic model is composed of two main components: a HPDP topic model for the text and hashtags, and a GP based random function model for the followers network. The authorship information serves to connect the two together.

We design the **HPDP topic model** as follows. For the word distributions, we first generate a parent word distribution prior $\gamma$ for all topics:

$$\gamma \sim \text{PDP}(\alpha^{\gamma}, \beta^{\gamma}, H_{\gamma}), \tag{1}$$

where $H_{\gamma}$ is the discrete uniform distribution over the word vocabulary $V$. Then, we sample the hashtag distributions $\psi'_k$ and word distributions $\psi_k$ for each topic $k$, treating $\gamma$ as the base distribution:

$$\psi'_k|\gamma \sim \text{PDP}(\alpha^{\psi'}, \beta^{\psi'}, \gamma), \tag{2}$$

$$\psi_k|\gamma \sim \text{PDP}(\alpha^{\psi}, \beta^{\psi}, \gamma). \tag{3}$$

Note that the tokens of hashtags are shared with the words, *i.e.* the hashtag *#happy* shares the same token as the word *happy*. This treatment is important as some hashtags are used as words instead of just labels, and it also allows any arbitrary words to be hashtags.

For topic distributions, we generate a global topic distribution $\mu_0$ that serves as a prior. Then generate the authors' topic distributions $\nu_i$ for each author $i$, and a miscellaneous topic distribution $\mu_1$ to capture topics that deviate from the authors' usual topics:

$$\mu_0 \sim \text{PDP}(\alpha^{\mu_0}, \beta^{\mu_0}), \tag{4}$$

$$\mu_1|\mu_0 \sim \text{PDP}(\alpha^{\mu_1}, \beta^{\mu_1}, \mu_0), \tag{5}$$

$$\nu_i|\mu_0 \sim \text{PDP}(\alpha^{\nu_i}, \beta^{\nu_i}, \mu_0). \tag{6}$$

For each tweet $m$, given the $\nu$'s and the observed authors $\mathbf{a}_m$, we sample the mixing proportions $\rho^{\nu}_m$ and the observed-authors topic distribution $\eta_m$:

$$\rho^{\nu}_m|\mathbf{a}_m \sim \text{Dir}(\lambda^{\nu}_1, \ldots, \lambda^{\nu}_{|\mathbf{a}_m|}), \tag{7}$$

$$\eta_m|\mathbf{a}_m, \rho^{\nu}_m, \nu \sim \text{PDP}(\alpha^{\eta}, \beta^{\eta}, \textstyle\sum_i \rho^{\nu}_{m,i}\nu_i). \tag{8}$$

The mixing proportions $\rho^{\nu}_m$ determine the contribution of each author in the document, although in the case of tweets, $|\mathbf{a}_m| = 1$ and hence $\rho^{\nu}_m = \{1\}$. Next, we generate the topic distributions of the observed hashtags ($\theta'_m$) and the observed words ($\theta_m$), following the technique in the adaptive topic model (Du et al., 2012). We explicitly model the influence of hashtags to words, by generating the

words conditioned on the hashtags. The intuition comes from hashtags being the themes of a tweet, and they drive the content of the tweet. Specifically, we sample the mixing proportions $\rho_m^{\theta'}$, which control the contribution of $\eta_m$ and $\mu_1$ in the base distribution of $\theta'_m$, and then generate $\theta'_m$:

$$\rho_m^{\theta'} \sim \text{Beta}(\lambda_0^{\theta'}, \lambda_1^{\theta'}), \tag{9}$$

$$\theta'_m|\mu_1, \eta_m \sim \text{PDP}(\alpha_m^{\theta'}, \beta_m^{\theta'}, \rho_m^{\theta'}\mu_1 + (1-\rho_m^{\theta'})\eta_m). \tag{10}$$

We set $\theta'_m$ and $\eta_m$ as the parent distributions of $\theta_m$. This flexible configuration allows us to investigate the relationship between $\theta_m$, $\theta'_m$ and $\eta_m$, *e.g.*, we can examine if $\theta_m$ is directly determined by $\eta_m$, or through the $\theta'_m$. The mixing proportions $\rho_m^{\theta}$ and $\theta_m$ is generated similarly:

$$\rho_m^{\theta} \sim \text{Beta}(\lambda_0^{\theta}, \lambda_1^{\theta}), \tag{11}$$

$$\theta_m|\eta_m, \theta'_m \sim \text{PDP}(\alpha_m^{\theta}, \beta_m^{\theta}, \rho_m^{\theta}\eta_m + (1-\rho_m^{\theta})\theta'_m). \tag{12}$$

The hashtags and words are then generated in a similar fashion as LDA. For each of the $N'_m$ hashtags, sample a topic and a hashtag:

$$z'_{m,n'}|\theta'_m \sim \text{Discrete}(\theta'_m), \tag{13}$$

$$y_{m,n'}|z'_{m,n'}, \psi'_{1:K} \sim \text{Discrete}(\psi'_{z'_{m,n'}}). \tag{14}$$

For each of the $N_m$ words, sample a topic and a word:

$$z_{m,n}|\theta_m \sim \text{Discrete}(\theta_m), \tag{15}$$

$$w_{m,n}|z_{m,n}, \psi_{1:K} \sim \text{Discrete}(\psi_{z_{m,n}}). \tag{16}$$

We note that all above $\alpha$'s, $\beta$'s and $\lambda$'s are hyperparameters of the model. Although the HPDP topic model may seem complex, it is actually a simple network of PDP nodes since all distributions on the probability vectors are modeled by the PDP.

At the network side, we adapt the **GP based random function model** (Lloyd et al., 2012) to model the followers network. The network modeling is connected to the HPDP topic model *via* the author topic distributions $\nu$'s, where we treat them as inputs to the GP in the network model. The GP, denoted as $\mathcal{F}$, is used in determining $x_{ij}$, the binary variables indicating the existence of the social links between the authors. For each pair of authors $(i, j)$, we sample their connections with the following random function model:

$$Q_{ij}|\nu_{1:A} \sim \mathcal{F}(\nu_i, \nu_j), \tag{17}$$

$$w_{ij}|Q_{ij} = \sigma(Q_{ij}), \tag{18}$$

$$x_{ij}|w_{ij} \sim \text{Bernoulli}(w_{ij}), \tag{19}$$

where $\sigma()$ is the *sigmoid function*, $\mathcal{F}$ is modeled as a GP.

## Appendix B Chinese Restaurant Process (CRP) Representation for the Twitter-Network Topic Model

For the **topic model**, we adopt a Chinese Restaurant Process (CRP) metaphor (Teh and Jordan, 2010; Blei et al., 2010) to represent the variables. We represent all words and hashtags as customers; the PDP distributed nodes as restaurants; and the topics as dishes. We will use these terms interchangeably in this paper, *e.g.*, *topic* $\equiv$ *dish*. The intuition behind this is straightforward: in each restaurant, each customer is allocated a table to sit at, and each table serves only one dish. Hence, customers (words) who are on the same table share the same dish (topic). This is similar to the 'counts' in LDA, albeit complicated by the fact that different tables can serve the same dish. Moreover, a table in a restaurant is treated as a customer in its parent restaurant. Below shows a detailed explanation on how this works.

For each document $m$, the first *word* enters *node* $\theta_m$ and opens a new table, which serves a dish (topic) $k$ that is available from its parent *nodes*. This *word* is assigned a topic $k$. Subsequent *words*, upon entering node $\theta_m$, can then choose to sit at the available tables or open a new table. If they choose to sit at the existing tables, the *word* will have the same dish as other *words* on the same table (each table serves only one dish). If a new table is opened, a new dish is sampled from the

parent *nodes*. The same process happens to *hashtags*, which enter node $\theta'_m$ instead of $\theta_m$. Note that all newly opened tables become a new customer in the parent *node* which offered the dish. This process repeats recursively (for all customers) until the root ($\mu_0$), in which opening a new table means inventing a new dish (creating a new topic) that can be passed down to children *nodes*. Note that for word distributions $\gamma$, $\psi'$ and $\psi$, the dishes are vocabulary words. The customers no longer choose table randomly since the dish corresponds to each customer is already known (words and hashtags are observed variables).

Naively recording the seating arrangements (table and dish) of each customer brings computational inefficiency in posterior inference. Instead, we marginalize the PDP and use the table multiplicity (or table counts) representation (Chen et al., 2011), which requires no dynamic memory, thus consumes only a factor of memory at no loss of inference efficiency. For each restaurant/node $\mathcal{N}$, we store $c_k^{\mathcal{N}}$, the number of customers having dish $k$, and $t_k^{\mathcal{N}\to\mathcal{P}}$, the number of tables serving dish $k$ from parent restaurant/node $\mathcal{P}$. For example, $c_k^{\theta_m}$ is the number of customers in restaurant $\theta_m$ (the number of words in document $m$ that is assigned topic $k$). For each node $\mathcal{N}$, we also define the total number of customers as $C^{\mathcal{N}} = \sum_k c_k^{\mathcal{N}}$, the total number of tables serving dish $k$ as $t_k^{\mathcal{N}} = \sum_{\mathcal{P}} t_k^{\mathcal{N}\to\mathcal{P}}$, the total number of tables serving dishes from node $\mathcal{P}$ as $T^{\mathcal{N}\to\mathcal{P}} = \sum_k t_k^{\mathcal{N}\to\mathcal{P}}$, and the number of total tables as $T^{\mathcal{N}} = \sum_k t_k^{\mathcal{N}} = \sum_{\mathcal{P}} T^{\mathcal{N}\to\mathcal{P}}$. Note that $c_k^{\mathcal{P}} = \sum_{\mathcal{N}} t_k^{\mathcal{N}\to\mathcal{P}}$ for all $\mathcal{P}$ except $\theta'_m$ and $\theta_m$.

After marginalizing out the latent variables, we can write down the model likelihood in terms of $c_k^{\mathcal{N}}$ and $t_k^{\mathcal{N}\to\mathcal{P}}$. We denote $\mathbf{W}$ and $\mathbf{Y}$ as the set of all words and tags; $\mathbf{Z}$ and $\mathbf{Z}'$ as the set of all topic assignments for words and tags; $\mathbf{T}$ as the set of all table multiplicities; and $\mathbf{\Phi}$ as the set of all model parameters (*e.g.* $\alpha$). The likelihood can easily be read out as $p_1(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{Z}', \mathbf{T}|\mathbf{\Phi}) \propto$

$$f(\mu_0)f(\mu_1)\Big(\prod_{i=1}^{A} f(\nu_i)\Big)\Big(\prod_{k=1}^{K} f(\psi'_k)f(\psi_k)\Big)f(\gamma)\Big(\prod_{m=1}^{M} f(\eta_m)f(\theta'_m)f(\theta_m)g(\rho_m^{\theta'})g(\rho_m^{\theta})\Big), \quad (20)$$

where $f(\mathcal{N})$ is the likelihood corresponding to node $\mathcal{N}$ and $g(\rho)$ is the likelihood corresponding to the probability $\rho$ that controls which parent node to send a customer to. These likelihoods have the following forms:

$$f(\mathcal{N}) = \frac{(\beta^{\mathcal{N}}|\alpha^{\mathcal{N}})_{T^{\mathcal{N}}}}{(\beta^{\mathcal{N}})_{C^{\mathcal{N}}}} \prod_k S^{c_k^{\mathcal{N}}}_{t_k^{\mathcal{N}},\alpha^{\mathcal{N}}} \ , \quad (21)$$

$$g(\rho^{\mathcal{N}}) = B(\lambda_0^{\mathcal{N}} + T^{\mathcal{N}\to\mathcal{P}_0}, \lambda_1^{\mathcal{N}} + T^{\mathcal{N}\to\mathcal{P}_1}). \quad (22)$$

$(x)_T$ and $(x|y)_T$ denote the Pochhammer symbol, $(x|y)_T = x(x+y)\dots(x+(T-1)y)$ and $(x)_T = (x|1)_T$. $S^c_{t,a}$ is the generalized Stirling number, see Buntine and Hutter (2012), and $B(x, y)$ denotes the Beta function that normalizes a Dirichlet. With the CRP presentation, the likelihood is modularized into product of nodes' likelihood, which allows the posterior to compute very quickly.

In the **network model**, the GP based random function $\mathcal{F}$ allows us to easily marginalize out the entries in $\mathcal{F}$ without observations, resulting in a finite dimension Gaussian prior. The conditional posterior is written as $p_2(\mathcal{X}, \{Q_{ij}\}|\{\nu_i\}) \propto$

$$\prod_i \prod_j \left(\sigma(Q_{ij})^{x_{ij}} \left(1-\sigma(Q_{ij})\right)^{1-x_{ij}}\right) |\kappa|^{-\frac{1}{2}} \times \exp\left(-\frac{1}{2}\left(\mathbf{Q}-\xi\right)^T \kappa^{-1} \left(\mathbf{Q}-\xi\right)\right), \quad (23)$$

where $Q \sim \text{GP}(\xi, \kappa)$, $\xi$ denotes the mean function and $\kappa$ is the kernel function in the GP. Following Lloyd et al. (2012), we would concatenate the author topic distributions ($\nu_{m1}$ and $\nu_{m2}$) as the feature for link $m$, and use them in the kernel function. However, this definition fails to consider the relation between author topic distributions, *i.e.*, we expect authors with similar topics are connected and *vice versa*. To overcome this, we propose a new kernel function $\kappa_{mn}(\epsilon_m, \epsilon_n) =$

$$\frac{s^2}{2}\exp\left(-\frac{|\text{Dist}(\nu_{m1},\nu_{m2}) - \text{Dist}(\nu_{n1},\nu_{n2})|^2}{2l^2}\right) + \sigma^2\delta(m=n), \quad (24)$$

where $s, l, \sigma$ are hyperparameters; $\epsilon_m = [\nu_{m1}, \nu_{m2}]$; $\delta(\cdot)$ is the indicator function; $\text{Dist}(\nu_{m1}, \nu_{m2})$ is an arbitrary distance function, we use the *cosine similarity* in this paper. We can see that if $\epsilon_m$ and $\epsilon_n$ have similar distance, they are likely to behave similarly. In addition, we set the mean function as $\xi(\epsilon_m) = \text{Dist}(\nu_{m1}, \nu_{m2})$.

## Appendix C Detailed Inference Procedure for the HPDP Topic Model

Combining a GP with a HPDP makes posterior inference for the TN topic model nontrivial. Hence, we employ approximate inference by alternatively perform Markov chain Monte Carlo (MCMC) sampling on the topic model and the network model, conditioned on each other. We develop a framework to perform collapse Gibbs sampling generally on any Bayesian network of PDPs, built upon the work of Buntine et al. (2010) and Chen et al. (2011), which allows quick prototyping and development of new variants of topic model. For the network model, we derive a Metropolis-Hastings (MH) algorithm based on the elliptical slice sampler (Murray et al., 2009). In addition, the author topic distributions ($\nu_i$) connecting the HPDP and GP are sampled with an MH schema since their posteriors do not follow a standard form.

### Appendix C.1 Collapsed Gibbs Sampling

Following Chen et al. (2011), we assign a Bernoulli variable $u$ to each customer to indicate whether the customer created the table, also known as the 'head' of the table. Doing so avoids the need to record all seating arrangements and also improves the algorithm considerably. The Gibbs sampling procedures follow standard LDA, *i.e.* for each word (and hashtag), decrement the observation, sample a new topic for the word and increment the associated counts; though each of the procedures is more complicated here. We note that our collapsed Gibbs sampler is general for any HPDP topic model represented by a PDP network. The only difference would be the explicit representation of the posterior likelihood and counts.

Here, we describe a straightforward Gibbs sampling algorithm for training the HPDP topic model (*i.e.* without network). The full conditional posterior probability for collapsed block Gibbs sampling can be derived easily. For instance, the conditional posterior in sampling the topic assignment of word $w_{m,n}$ is

$$p(z_{m,n}, \mathbf{T}|\mathbf{W}, \mathbf{Y}, \mathbf{Z}^{-m,n}, \mathbf{Z}', \mathbf{T}^{-m,n}, \mathbf{\Phi}) = \frac{p_1(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{Z}', \mathbf{T}|\mathbf{\Phi})}{p_1(\mathbf{W}, \mathbf{Y}, \mathbf{Z}^{-m,n}, \mathbf{Z}', \mathbf{T}^{-m,n}|\mathbf{\Phi})} \tag{25}$$

where the superscript $^{-m,n}$ indicates the word $w_{m,n}$ is removed from the respective sets. This ratio is easy to compute because the table multiplicity $t_k^{\mathcal{N}}$ and the customer counts $c_k^{\mathcal{N}}$ will only increment by at most 1, allowing simplification of the ratio of Pochhammer symbol and Beta function. The ratio of Stirling number can be computed quickly via caching (see Buntine and Hutter (2012)). Similarly, the conditional posterior probability for sampling the topic assignments of hashtag $y_{m,n'}$ can be derived as

$$p(z'_{m,n'}, \mathbf{T}|\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{Z}'^{-m,n'}, \mathbf{T}^{-m,n'}, \mathbf{\Phi}) = \frac{p_1(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{Z}', \mathbf{T}|\mathbf{\Phi})}{p_1(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{Z}'^{-m,n'}, \mathbf{T}^{-m,n'}|\mathbf{\Phi})} \tag{26}$$

**Decrementing a Word or a Hashtag** To remove a word or a hashtag to perform Gibbs sampling, we introduce an auxiliary variable similar to table indicator (Chen et al., 2011). Note that our table indicator representation is different to that of Chen et al. (2011), due to the complexity of the TN topic model. Instead of having a variable which indicates the level of table contribution, our table indicators show to which parents a node is contributing a table. The sample space of the indicator of a node is its parent nodes $\mathcal{P}_1, \ldots, \mathcal{P}_P$, plus the empty set $\emptyset$.

When a customer (a hashtag or a word) having dish $k$ is removed from node $\mathcal{N}$, we sample an indicator $u_k^{\mathcal{N}}$, which indicates whether to remove a table serving dish $k$ and from which parent nodes. When $u_k^{\mathcal{N}}$ is equal to $\mathcal{P}_i$, we remove a table serving dish $k$ from node $\mathcal{P}_i$, decrement $t_k^{\mathcal{N}\to\mathcal{P}_i}$ and recursively remove a customer in node $\mathcal{P}_i$ (since the table removed is a customer in node $\mathcal{P}_i$). We repeat the process recursively until the root node is reached, or until $u_k^{\mathcal{N}}$ equals $\emptyset$, which means the customer does not contribute to any table.

The value of $u_k^{\mathcal{N}}$ is sampled as follows:

$$p(u_k^{\mathcal{N}}) = \begin{cases} t_k^{\mathcal{N}\to\mathcal{P}_i}/c_k^{\mathcal{N}} & \text{if } u_k^{\mathcal{N}} = \mathcal{P}_i \\ 1 - \sum_{\mathcal{P}_i} p(u_k^{\mathcal{N}} = \mathcal{P}_i) & \text{if } u_k^{\mathcal{N}} = \emptyset \end{cases} \tag{27}$$

We give an illustrative example: When a word $w_{m,n}$ (with topic $z_{m,n}$) is removed, we decrement $c^{\theta_m}_{z_{m,n}}$, *i.e.* $c^{\theta_m}_{z_{m,n}} = c^{\theta_m}_{z_{m,n}} - 1$. Then we determine if this word contributes to any table in node $\theta_m$, by sampling $u^{\theta_m}_{z_{m,n}}$, if $u^{\theta_m}_{z_{m,n}} = \emptyset$, we do not remove any table and proceed with the next step in the Gibbs sampling; otherwise, $u^{\theta_m}_{z_{m,n}}$ can be either $\theta'_m$ or $\eta_m$, in these cases, we would decrement $t^{\theta_m \to u^{\theta_m}_{z_{m,n}}}_{z_{m,n}}$ and $c^{u^{\theta_m}_{z_{m,n}}}_{z_{m,n}}$, and continue the process recursively.

**Sampling** The Gibbs sampling procedures follow standard LDA, *i.e.* for each word (and hashtag), decrement the observation, sample a new topic for the word and increment the associated counts; though each of the procedures is more complicated here. The algorithm for Gibbs sampling is summarized in Algorithm 1.

**Algorithm 1** Collapsed Gibbs Sampling for the HPDP Topic Model

1. Initialize the model by assigning topics to each word and each hashtag randomly, building the relevant customer counts $c^{\mathcal{N}}_k$ and table counts $t^{\mathcal{N}}_k$.
2. For each document $m$:
   (a) For each word $w_{m,n}$:
      i. Remove the word and decrement associated counts.
      ii. Sample $z_{m,n}$ and $\mathbf{T}$ from the conditional posterior (Equation 25).
      iii. Increment the associated counts for the sampled topic.
   (b) For each hashtag $y_{m,n'}$:
      i. Remove the hashtag and decrement associated counts.
      ii. Sample $z'_{m,n'}$ and $\mathbf{T}$ from the conditional posterior (Equation 26).
      iii. Increment the associated counts for the sampled topic.
3. Repeat step 2 until the model converges or when a fix number of iteration is reached.

**Elliptical Slice Sampling and MH Update** We jointly sample the author topic distribution $\nu_i$ associated with the GP with an MH procedure. Specifically, we use independent Dirichlet distributed proposals $q(\nu'_i|\nu_i) = q(\nu'_i)$, which are the posteriors of $\nu_i$ without the GP likelihood. The MH sampling procedure is summarized in Algorithm 2.

**Algorithm 2** The Metropolis-Hastings Algorithm

1. Propose a new $\nu'_i$ from its proposal distribution
$$q(\nu'_i|\nu_i) = \text{Dir}(c^{\nu_i}_1 - \alpha^{\nu_i} T^{\nu_i}_1 + (\alpha^{\nu_i} T^{\nu_i} + \beta^{\nu_i})\mu_{0,1}, \cdots, c^{\nu_i}_K - \alpha^{\nu_i} T^{\nu_i}_K + (\alpha^{\nu_i} T^{\nu_i} + \beta^{\nu_i})\mu_{0,K}).$$
2. Re-sample $Q_{ij}$ with the elliptical slice sampler, using $\xi'$ and $\kappa'$ calculated from $\nu'_i$.
3. Accept the proposed $\nu'_i$ with probability $A = \min\left\{1, \frac{p_2(\mathcal{X},\{Q_{ij}\}|\{\nu'_i\})}{p_2(\mathcal{X},\{Q_{ij}\}|\{\nu_i\})} \frac{\text{CRP}(\{\eta_m\},\nu'_i)}{\text{CRP}(\{\eta_m\},\nu_i)} \frac{q(\nu_i)}{q(\nu'_i)}\right\}$, where $\text{CRP}(\{\eta_m\},\nu_i)$ denotes the probability of simulating the CRP for each $\eta_m$ with base distribution $\nu_i$.

**Hyperparameter Sampling** We sample the hyperparameters $\beta$ using the *auxiliary variable sampler* following Teh (2006). Sampling the concentration parameters allows the topic distributions of each author to vary, *i.e.* some authors have few very specific topics and some other authors can have many different topics, which is very important.

While both the $\alpha$ and $\beta$ can be sampled with the auxiliary variable sampler, it is more efficient by fixing $\alpha$ and sample only $\beta$. We also found the values of the parameters $s$, $l$ and $\sigma$ have no significant impact on the model performance, thus we fixed them to 1 in the experiments.

Assuming $\beta$ has a Gamma distributed hyperprior with shape $a$ and rate $b$, we sample a new parameter $\beta'$ of node $\mathcal{N}$ as follows:

1. Sample $x \sim Beta(C^{\mathcal{N}}, \beta)$.
2. For $i$ from 0 to $(T^{\mathcal{N}} - 1)$, sample $y_i \sim Bernoulli(\beta/(\beta + i\alpha))$.
3. Sample new $\beta' \sim Gamma(a + \sum_i y_i, b - \log(1 - x))$.

**Posterior Inference** The final topic distributions and word distributions are reconstructed from the table counts and customer counts. More specifically, the topic distribution (or word distribution) $\mathbf{s} = \{s_0, \ldots, s_K\}$ of each node $\mathcal{N}$ is estimated from its posterior mean:

$$E(s_k|\cdot) = \frac{c_k^{\mathcal{N}} - \alpha^{\mathcal{N}} t_k^{\mathcal{N}} + (\alpha^{\mathcal{N}} T^{\mathcal{N}} + \beta^{\mathcal{N}}) p_k}{C^{\mathcal{N}} + \beta^{\mathcal{N}}}$$

where $p_k$ is the base distribution (the parents' distribution) and $K$ is the number of seen dishes. For instance, the topic distribution $\nu_i$ of each author $i$ can be calculated recursively by first calculating the topic distribution of $\mu_0$. The word distribution of each topic can be computed similarly.

Additionally, we can perform posterior predictive inference for various applications, such as author recommendation by predicting the followers network.

**Author Recommendation** The goal of author recommendation is to predict the most likely authors in the training dataset a new author/user would connect to. This is straightforward within the GP framework. Suppose there are $A$ authors in the training dataset, given a new author $j$, we first infer its *author topic distribution* $\nu_j$ using the trained *topic model*. Based on results from the Gaussian process regression (Rasmussen and Williams, 2006), the strength of the link between the new author $j$ and author $i$ in the training dataset (quantified by $Q_{ij}$) is a Gaussian random variable with mean $\tilde{Q}_{\epsilon_*^i} = \kappa(\epsilon_*^i, \epsilon_{1:A}^i)\kappa(\epsilon_{1:A}^i, \epsilon_{1:A}^i)^{-1} Q_{i,1:A}$ and covariance $\tilde{\kappa} = \kappa(\epsilon_*^i, \epsilon_*^i) - \kappa(\epsilon_*^i, \epsilon_{1:A}^i)\kappa(\epsilon_{1:A}^i, \epsilon_{1:A}^i)^{-1}\kappa(\epsilon_{1:A}^i, \epsilon_*^i)$, where $\epsilon_*^i = [\nu_i; \nu_j]$ and $\epsilon_t^i = [\nu_i; \nu_t]$. Author recommendation can thus be done by choosing $n$ authors with the largest $\tilde{Q}_{\epsilon_*^i}$ values as recommended authors for the new user $j$.

## Appendix D Additional Results

### Appendix D.1 Clustering and Topic Coherence

Mehrotra et al. (2013) shows that using LDA on pooled tweets[1] gives significant improvement. Here, in contrast to their *ad-hoc* technique, we demonstrate that we can achieve a significantly better performance with the TN topic model. We evaluate the TN topic model with standard clustering measures, *i.e.* purity and normalized mutual information (NMI) (Manning et al., 2008), and topic coherence measured by pointwise mutual information (PMI) (Newman et al., 2009). Note that due to the lack of network information in this dataset, the TN topic model is equivalent to the ablated model without network. Following Mehrotra et al. (2013), we assign each tweet to a cluster based on its most dominant topic, and compare them with the ground truth, which are tweets queried with certain keywords such as 'movie' and 'food'. Since purity can be trivially improved with the number of clusters, we limit the maximum number of topics to 20 for a fair comparison. We present the results in the last row of Table 1, other rows are different pooling methods used with LDA, obtained from the paper by Mehrotra et al. (2013) (Table 4). We can see that the TN topic model outperforms all other methods.

[1] Multiple tweets combined into a single document.

Table 1: Clustering and Topic Coherence Results

| Methods | Purity | | | NMI Score | | | PMI score | | |
|---|---|---|---|---|---|---|---|---|---|
| | Generic | Specific | Events | Generic | Specific | Events | Generic | Specific | Events |
| No pooling | 0.49 | 0.64 | 0.69 | 0.28 | 0.22 | 0.39 | −1.27 | 0.47 | 0.47 |
| Author | 0.54 | 0.62 | 0.60 | 0.24 | 0.17 | 0.41 | 0.21 | 0.79 | 0.51 |
| Hourly | 0.45 | 0.61 | 0.61 | 0.07 | 0.09 | 0.32 | −1.31 | 0.87 | 0.22 |
| Burstwise | 0.42 | 0.60 | 0.64 | 0.18 | 0.16 | 0.33 | 0.48 | 0.74 | 0.58 |
| Hashtag | 0.54 | **0.68** | 0.71 | 0.28 | 0.23 | 0.42 | 0.78 | **1.43** | 1.07 |
| TN | **0.66** | **0.68** | **0.79** | **0.43** | **0.31** | **0.52** | **0.79** | 0.81 | **1.66** |

### Appendix D.2 Inference on the Mixing Probabilities

The posterior of the mixing probability $\rho^{\mathcal{N}}$ gives insight on the influence of the parents' distributions of a distribution node $\mathcal{N}$. The posterior mean of $\rho^{\mathcal{N}}$ can be computed as:

$$E(\rho^{\mathcal{N}}|\cdot) = \frac{T^{\mathcal{N}\to\mathcal{P}_0} + \lambda_0^{\mathcal{N}}}{T^{\mathcal{N}\to\mathcal{P}_0} + \lambda_0^{\mathcal{N}} + T^{\mathcal{N}\to\mathcal{P}_1} + \lambda_1^{\mathcal{N}}}$$

where $\mathcal{P}_0$ and $\mathcal{P}_1$ are the first and second parent nodes of node $\mathcal{N}$, and $\rho^{\mathcal{N}}$ is the proportion of influence of the first parent. More specifically, $\rho_m^{\theta'}$ is the proportion of the influence of the miscellaneous topic distribution (node $\mu_1$) to node $\theta'_m$, and $1 - \rho_m^{\theta'}$ is the proportion of influence from the author's topic distribution.

In the TN topic model, we expect that $\rho_m^{\theta'}$ to be small since the miscellaneous topic distribution is designed to capture topics that are not frequently used by authors. Figure 1 displays the empirical cumulative frequency plot for the posterior of $\rho_m^{\theta'}$. We can see that more than 80% of the estimated $\rho_m^{\theta'}$ have proportion less than 0.4, which confirms that the TN topic model is working as intended.

On the other hand, the mixing probability $\rho_m^{\theta}$ is inversely related to the influence of hashtags to the words, *i.e.* the lower the value of $\rho_m^{\theta}$, the higher their influence. Figure 2 shows a relatively linear empirical cumulative frequency plot, suggesting that equal contribution of $\eta_m$ and $\theta'_m$ towards node $\theta_m$. This mean that the hashtags do influence the words to a great extent.

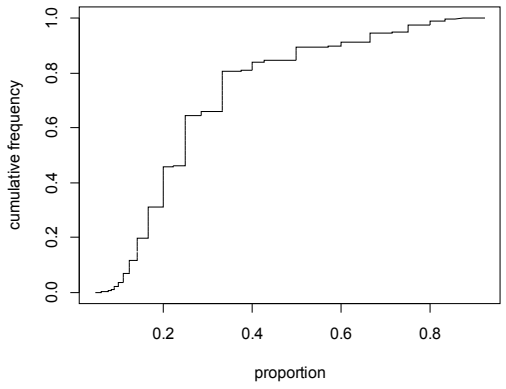


Figure 1: Cumulative Frequency of $\rho_m^{\theta'}$

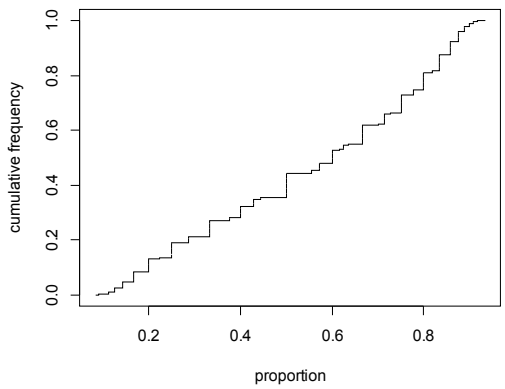


Figure 2: Cumulative Frequency of $\rho_m^{\theta}$

### Appendix D.3 Additional Topic Explorations

In this subsection, we show the qualitative results by running topic exploration task with the TN topic model. Table 2 shows the top 10 significant topics from our corpus of 60370 tweets; while Table 3, 4, and 5 show the top 10 topics from the dataset of Mehrotra et al. (2013).

Table 2: Topics for 60370 Tweets

| | Hashtags | Words |
|---|---|---|
| T0 | #finance #money #economy | finance money bank marketwatch stocks china group shares |
| T1 | #politics #iranelection #tcot | politics iran iranelection tcot tlot topprog obama music |
| T2 | #music #folk #pop | music folk monster head pop free indie album gratuit dernier |
| T3 | #music #techno #listening | music alexanderfog techno video gardian listening nyt videos |
| T4 | #sports #women #asheville | sports women football win game top world asheville vols team |
| T5 | #tech #news #jobs | tech news jquery jobs hiring engineer gizmos google reuters |
| T6 | #politics #news #sonnet | politics found news obama health care sonnet open david palin |
| T7 | #politics #news #actavist | politics news actavist wales bbc welshassembly iamcrazydave |
| T8 | #science #news #biology | science news source study scientists cancer researchers brain biology health |
| T9 | #tech #web #technology | web tech technology found news social google iphone twitter |

Table 3: Topics for "Specific" Dataset

| | Tags | Words |
|---|---|---|
| T0 | #obama #tcot #news | obama president barack news michelle white house post usa |
| T1 | #iphone #apple #app | apple iphone store app at&t free mac check update itunes |
| T2 | #pakcricket #cricket #baseball | usa baseball cricket game team susan france spain cup brazil |
| T3 | #microsoft #usability #bing | microsoft thousands followers free windows bing found search |
| T4 | #iranelection #gr88 #mousavi | mousavi iranelection iran tehran gr88 arrested rally supporters |
| T5 | #obama #lgbt #doma | obama health president care bill reform gay barackobama plan |
| T6 | #ipod #iphone #twitools | apple ipod touch xbox sale microsoft iphone mp3 game |
| T7 | #iranelection #neda #mousavi | neda mousavi iran iranelection tehran internet anonymous bypass send blocking |
| T8 | #jobs #tweetmyjobs #twitteranalyzer | united states jobs usa job manager thousands france business services |
| T9 | #iranelection #iran #gr88 | obama iran iranelection mousavi iranian election ahmadinejad president people tehran |

Table 4: Topics for "Generic" Dataset

| | Tags | Words |
|---|---|---|
| T0 | #ecademy #socialmedia #twitter | business home marketing internet online social based blog twitter media |
| T1 | #tcot #healthcare #gop | health care obama reform insurance plan public tcot news |
| T2 | #swineflu #h1n1 #thcommandment | health flu swine united states news world officials h1n1 |
| T3 | #food #cook #whereisgraeme | food good eat fast great family wine eating fun network |
| T4 | #iranelection #neda #gr88 | family iranelection iran neda design sport iranian home business tehran |
| T5 | #news #business #mydonut | business design news small blog social media health plan |
| T6 | #design #webdesign #wordpress | design web blog website graphic logo business inspiration site wordpress |
| T7 | #health #foodinc #food | health food blog fitness weight diet healthy post good news |
| T8 | #business #loan #quickbooks | business small free online home cards money make start blog |
| T9 | #travel #contest #tuesgiveaway | family fun food free summer movie house sale single june |

Table 5: Topics for "Events" Dataset

| | Tags | Words |
|---|---|---|
| T0 | #conf #openvideo #conference | conference press live news call video twitter great today june |
| T1 | #swineflu #swine #health | flu swine cases news death health confirmed pandemic case |
| T2 | #recession #jobs #tweetmyjobs | recession business money jobs jacksonville proof make home marketing great |
| T3 | #nba #lakers #jtv | lakers game magic nba finals let0̃027s los orlando angeles day |
| T4 | #iranelection #iran #tehran | iran election iranelection elections tehran attack attacks twitter protests ahmadinejad |
| T5 | #michaeljackson #michael #nfl | jackson michael hospital cardiac arrest breakingnews los angeles bulletin reports |
| T6 | #pakcricket #cricket #pakistan | world pakistan live cricket cup india june sri icc lanka |
| T7 | #sanford #tcot #fare | flight air france conference breakingnews madoff world0̆027s fraud prison sentenced |
| T8 | #lakers #nba #ilist | lakers jackson phil nba coach kobe fined bryant los title |
| T9 | #tcot #news #gop | scandal attack attacks sex scandals news attacked tcot cheney |

## References


D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *JACM*, 57(2):7, 2010.

W. Buntine and M. Hutter. A Bayesian review of the Poisson-Dirichlet process. *arXiv preprint arXiv:1007.0296v2*, 2012.

W. Buntine, L. Du, and P. Nurmi. Bayesian networks on Dirichlet distributed vectors. *Proceedings of the fifth European workshop on probabilistic graphical models*, 2010.

C. Chen, L. Du, and W. Buntine. Sampling table configurations for the hierarchical Poisson-Dirichlet process. In *ECML*. 2011.

L. Du, W. Buntine, and H. Jin. Modeling sequential text with an adaptive topic model. In *EMNLP-CoNLL*, 2012.

J. Lloyd, P. Orbanz, Z. Ghahramani, and D. Roy. Random function priors for exchangeable arrays with applications to graphs and relational data. In *NIPS*, 2012.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In *SIGIR*, 2013.

I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical slice sampling. In *AISTATS*, 2009.

D. Newman, S. Karimi, and L. Cavedon. External evaluation of topic models. In *ADCS*, 2009.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for Machine Learning*. MIT Press, 2006.

Y. W. Teh. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, NUS, 2006.

Y. W. Teh and M. I. Jordan. Hierarchical Bayesian nonparametric models with applications. In *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.