Suplementary Materials for "A Network Model for Dynamic Textual Communications with Application to Government Email Corpora"

Anonymous Authors[1]

1. Normalizing constant of Gibbs measure

The non-empty Gibbs measure (Fellows & Handcock, 2017) defines the probability of author a selecting the binary recipient vector $\boldsymbol{u}_{ad}$ as

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

To use this distribution efficiently, we derive a closed-form expression for $Z(\delta, \boldsymbol{\lambda}_{id})$ 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$(\delta + \lambda_{adr})$, then

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

This is straightforward to verify by looking at

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

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

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

Now, since

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

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

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

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:

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

Then, we have

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

Finally, the closed form expression for the normalizing constant under the non-empty Gibbs measure is

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

2. Psuedocode for inference

In Section 2, we outlined a Metropolis-within-Gibbs sampling algorithm and each latent variables conditional posterior. Algorithm 1 provides the pseudocod for the IPTM's inference. For every outer iteration o, we sequentailly resampling the value of each parameter from its conditional posterior given the observed data, hyperparamters, and the current values of the other parameters. For hyperparameter optimization, we fix $n_1 = 5$. Note that we specify a larger number of inner iterations for (n_2, n_3, and n_4) such as 5, because those variables require Metropolis-Hastings update which mixes slower than Gibbs update.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Algorithm 1 Markov Chain Monte Carlo (MCMC)

Input: data $\{(a_d, \boldsymbol{r}_d, t_d, \boldsymbol{w}_d)\}_{d=1}^{D}$,
number of interaction patterns and topics (C, K),
hyperparameters $(\alpha, \beta, \boldsymbol{m}, \boldsymbol{\mu}_b, \Sigma_b, \boldsymbol{\mu}_\eta, \Sigma_\eta, \mu_\delta, \sigma^2_\delta)$,
number of iterations (O, n_1, n_2, n_3, n_4)

```
Set initial values
for o = 1 to O do
  for n = 1 to n_1 do
    Optimize α and m using (Wallach, 2008)
  end for
  for d = 1 to D do
    for a ∈ [A]_{\a_d} do
      for r ∈ [A]_{\a} do
        Draw u_{adr} from Equation (13)
      end for
    end for
  end for
  for k = 1 to K do
    Draw l_k from Equation (14)
  end for
  for d = 1 to D do
    for n = 1to N_d do
      Draw z_{dn} from Equation (15)
    end for
  end for
  for n = 1 to n_2 do
    Draw b and δ from Equation (16)
  end for
  for n = 1to n_3 do
    Draw η from Equation (17)
  end for
  for n = 1 to n_4 do
    Draw σ_τ^2 from Equation (17)
  end for
end for
```

3. Psuedocode for posterior predictive checks

Algorithm 2 Generate new data for PPC

Input: number of new data to generate R,
observed text data $\{\boldsymbol{w}_d\}_{d=1}^{D}$,
estimated latent variables $(\boldsymbol{u}, \boldsymbol{l}, \boldsymbol{z}, \boldsymbol{b}, \delta, \boldsymbol{\eta}, \sigma^2_\tau)$,
hyperparameters $(\alpha, \beta, \boldsymbol{m})$,
number of vocabularies V

```
for r = 1 to R do
  Initialize N_{vk} and N_k from z and w
  for d = 1 to D do
    if N_d > 0 then
      for n = 1 to N_d do
        Draw w_{dn} from P(w_{dn} = v) = (N_{v z_{dn}} + β/V) / (N_{z_{dn}} + β)
        Increment N_{w_{dn} z_{dn}} and N_{z_{dn}}
      end for
    end if
    Compute x_d given {(a_d, r_d, t_d)}_{[1:(d−1)]}
    Draw (a_d, r_d, t_d) following Section 2.3
  end for
  Store every r^{th} new data {(a_d, r_d, t_d, w_d)}_{d=1}^{D}
end for
```

References

Fellows, Ian and Handcock, Mark. Removing phase transitions from gibbs measures. In *Artificial Intelligence and Statistics*, pp. 289–297, 2017.

Wallach, Hanna Megan. *Structured topic models for language*. PhD thesis, University of Cambridge, 2008.