Gibbs Update for Bayesian Degree Corrected Stochastic Block Model

February 10, 2017

The purpose of this document is to derive the update rule for the collapsed Gibbs sampler for the Bayesian DCSBM. That is, our goal is to find

$$\mathrm{P}(z_i = k \mid A, z_{\setminus i}).$$

In fact, we actually only need this expression up to a normalization constant that does not depend on k. It will also ultimately be computationally convenient to work with logarithms for the usual numerical stability reasons. That is, we're looking for an expression $q(A, z_{\setminus i}, k)$ such that

$$\log \mathrm{P}(z_i = k \mid A, z_{\setminus i}) = q(A, z_{\setminus i}, k) + c(A, z_{\setminus i}).$$

1 The Model

Borrowing notation from the Bayesian DCSBM, the basic setup is a model for multi-graphs:

$$\begin{aligned}
z &\sim \mathrm{Diri}(\alpha) \\
(\phi_i)_{z_i=l} &\sim \mathrm{Diri}(\gamma 1_{(n_l)}) \\
\theta_i &= n_{z_i}\phi_i \\
\eta_{lm} &\sim \mathrm{Gamma}(\kappa, \lambda) \\
A_{ij} \mid \theta, \eta, z &\overset{ind}{\sim} \mathrm{Poi}(\eta_{z_i z_j}\theta_i\theta_j),\ i \neq j \\
A_{ii} \mid \theta, \eta, z &\overset{ind}{\sim} \mathrm{Poi}(\frac{1}{2}\eta_{z_i z_i}\theta_i^2).
\end{aligned}$$

Note that this differs from the infinite Bayesian DCSBM in that we've moved to a fixed number of communities (Chinese restaraunt process replaced by Dirichlet distribution); this doesn't really matter much — it will turn out that the Gibbs sampler update can be trivially adapted back to the infinite model. Note that the generative model conditional on z, ϕ, η can equivalently be written as:

1. $E_{lm} \mid \eta_{lm}, z \ \sim\ \mathrm{Poi}(n_l n_m \eta_{lm})$ for $l \neq m$

2. $E_{ll} \mid \eta_{ll}, z \;\sim\; \mathrm{Poi}(n_l^2 \eta_{ll}/2)$

3. $(D_i + A_{ii})_{z_i=l} \mid \phi, z, \sum_m E_{lm} + E_{ll} \;\sim\; \mathrm{Multi}(\sum_m E_{lm} + E_{ll}, (\phi_i)_{z_i=l})$

4. Assign termini uniformly at random consistent with the counts in 4

Basically, we can view this model as first generating edges between communities and then, conditional on those counts, distributing the edge-terminis among the vertices within the community (the later implicitly exploiting the relationship between the Poisson and Dirichlet distributions).

2 Gibbs Update 1

The first step is

$$\mathrm{P}(z_i = k \mid A, z_{\setminus i}) \propto \mathrm{P}(A \mid z_{\setminus i}, z_i = k)\mathrm{P}(z_i = k \mid z_{\setminus i}).$$

The second term is a straightforward Dirichlet-Multinomial calculation, so our focus will be on the likelihood.

Symbol	Meaning
d_j	degree of vertex j
$d_{j\to m}$	number of edges of vertex j to community m
e_{lm}	number of edges between communities l and m
n_l	number of vertices in community l
n_{lm}	$\begin{cases} n_l n_m & l \neq m \\ n_l^2/2 & l = m \end{cases}$

Table 1: Notation for summary stats.

After repeated failed attempts to derive a smarter update, lets just go with the one from the original paper:

$$\begin{aligned}\mathrm{P}(A \mid z) \propto \prod_{l \leq m} & G(\kappa, \lambda)^{-1} G(e_{lm} + \kappa, n_{lm} + \lambda) \\ & \cdot \prod_l \frac{B(\gamma 1_{(n_l)} + (d_i + A_{ii})_{i:z_i=l})}{B(\gamma 1_{(n_l)})} n_l^{\sum_m e_{lm} + e_{ll}}.\end{aligned}$$

The basic update scheme works as follows:

1. Remove vertex v from the graph

2. Compute $q(k) = \log \mathrm{P}(A_{\setminus v}, A_v \mid z_{\setminus v}, z_v = k) + \mathrm{const}$ for each community k

3. Compute $r(k) = \log \mathrm{P}(z_v = k \mid z_{\setminus v})$

4. Sample community identity of v according to distribution $\mathrm{softmax}(q+r)$

5. Add vertex v back to the graph.

The idea is to write update equations that depend on sufficient stats of $A_{\setminus v}$ that can be cheaply updated in the vertex removal and addition steps.

To that end, taking the sufficient stats over $A_{\setminus v}$:

$$
\begin{aligned}
\mathrm{P}(A_{\setminus v}, A_v \mid z_{\setminus v}, z_v = k) \propto & \prod_{l \leq m : l \neq k, m \neq k} G(\kappa, \lambda)^{-1} G(e_{lm} + \kappa, n_{lm} + \lambda) \\
& \cdot \prod_{m \neq k} G(\kappa, \lambda)^{-1} G(e_{km} + d_v^{(m)}, n_{km} + \lambda + n_m) \\
& \cdot G(\kappa, \lambda)^{-1} G(e_{kk} + d_v^{(k)}, n_{kk} + \lambda + n_k + \frac{1}{2}) \\
& \cdot \prod_{m \neq k} \frac{B(\gamma 1_{(n_m)} + (d_i + A_{ii} + A_{vi})_{i:z_i = m})}{B(\gamma 1_{(n_m)})} n_m^{\sum_l e_{ml} + e_{mm} + d_v^{(l)}} \\
& \cdot \frac{B(\gamma 1_{(n_k+1)} + ((d_i + A_{ii} + A_{vi})_{i:z_i = k}, d_v + A_{vv}))}{B(\gamma 1_{(n_k+1)})} (n_k + 1)^{\sum_l e_{kl} + e_{kk} + d_v + d_v^{(k)} + A_{vv}} \\
= & \prod_{l \leq m} G(\kappa, \lambda)^{-1} G(e_{lm} + \kappa, n_{lm} + \lambda) \\
& \cdot \prod_{m} \frac{B(\gamma 1_{(n_m)} + (d_i + A_{ii} + A_{vi})_{i:z_i = m})}{B(\gamma 1_{(n_m)})} \\
& \cdot \prod n_m^{\sum_l e_{ml} + e_{mm} + d_v^{(l)}} \\
& \cdot \prod_{m \neq k} G(e_{km} + \kappa + d_v^{(m)}, n_{km} + \lambda + n_m) / G(e_{km} + \kappa, n_{km} + \lambda) \\
& \cdot G(e_{kk} + \kappa + d_v^{(k)} + A_{vv}, n_{kk} + \lambda + n_k + \frac{1}{2}) / G(e_{kk} + \kappa, n_{kk} + \lambda) \\
& \cdot \frac{B(\gamma 1_{(n_k+1)} + ((d_i + A_{ii} + A_{vi})_{i:z_i = k}, d_v + A_{vv}))}{B(\gamma 1_{(n_k)} + (d_i + A_{ii} + A_{vi})_{i:z_i = k})} \frac{B(\gamma 1_{(n_k)})}{B(\gamma 1_{(n_k+1)})} \\
& \cdot (n_k + 1)^{\sum_l e_{kl} + e_{kk} + d_v^{(k)} + d_v + A_{vv}} / n_k^{\sum_l e_{lk} + e_{kk} + d_v^{(k)}}
\end{aligned}
$$

the point being that the first three terms are free of k and so do not carry any information about the update, and the remaining terms are relatively simply functions of the hyper parameters and sufficient stats of the graph

Next,

$$
\begin{aligned}
GD(x, y, k, l) := \log \frac{G(x + k, y + l)}{G(x, y)} &= \log \frac{(y + l)^{-(x+k)} \Gamma(x + k)}{y^{-x} \Gamma(x)} \\
&= -(x + k) \log(y + l) + x \log y + \log \frac{\Gamma(x + k)}{\Gamma(x)}.
\end{aligned}
$$

Similarly, define

$$BD(x, y) := \log \frac{B((x, y))}{B(x)} = \log \Gamma(y) - \log \frac{\Gamma(\sum_i x_i + y)}{\Gamma(\sum_i x_i)}$$

and

$$BD(\emptyset, y) = 0.$$

In totality,

$$\begin{aligned}\log \mathrm{P}(A, A_v \mid z_{\setminus v}, z_v = k) = \mathrm{const} \\ &+ \sum_{m \neq k} GD(e_{km} + \kappa, n_{km} + \lambda, d_v^{(m)}, n_m) \\ &+ GD(e_{kk} + \kappa, n_{kk} + \lambda, d_v^{(k)} + A_{vv}, n_k + \frac{1}{2}) \\ &+ BD(\gamma 1_{(n_k)} + (d_i + A_{ii} + A_{vi})_{i:z_i = k}, d_v + A_{vv} + \gamma) \\ &- BD(\gamma 1_{(n_k)}, \gamma) \\ &+ (d_v + A_{vv}) \log(n_k + 1) \\ &+ (\sum_l e_{kl} + e_{kk} + d_v^{(k)}) \log(1 + \frac{1}{n_k}).\end{aligned}$$