# A Supplementary Material

## A.1 Experimental Settings

For all our experiments, we fix the variance across node community memberships $\alpha = 1$ and set our hyperparameters for $w_k$ to $\tau_a = 10$ and $\tau_b = 1$ across communities. We set an agressive learning rate so that $\mu_0 = 1$ and $\kappa = .5$. We use a restricted stratified node-sampling technique for all our experiments with the non-link partition set $m = 10$, unless stated otherwise. All experiments were run for 250,000 iterations from 5 random initializations with 10% of the links randomly held out along with an equal amount of non-links for testing. For the aMMSB, we used the same settings. The aMMSB uses a random initialization for $\theta_{ik} \sim \text{Gam}(100, .01)$ with hyperparameters over $w_k$ set to the expected number of link/non-links across $K$ uniformly distributed communities. The learning rate was set to $\mu_0 = 1024$ and $\kappa = .5$. We found these settings gave the best advantage for the aMMSB on these datasets that were optimized for its original experiments, with the exception of changing the Dirichlet prior to be uniform over its mixed-membership distributions ($\alpha = 1$), which we found to improve convergence for the aMMSB across our experiments.

## A.2 aHDPR ELBO

A more detailed representation of our ELBO for the aHDPR model can be seen here. Note that since we do not estimate $\phi_{ijk\ell}$, the ELBO needs to be computed in a more efficient manner:

$$\mathcal{L}(q) = \sum_{ij}^{E}\sum_{k=1}^{K}\left[\phi_{ijkk}\log f(w_k, y_{ij})\right] + \sum_{ij}^{E}\left[1 - (\sum_{k=1}^{K}\phi_{ijkk})\log f(\epsilon, y_{ij})\right] \tag{1}$$

$$+ \sum_{ij}^{E}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\left[\phi_{ijk\ell}(\mathbb{E}_q[\log(\pi_{ik})] + \mathbb{E}_q[\log(\pi_{j\ell})])\right] \tag{2}$$

$$+ \sum_{k=1}^{K}(\gamma - 1)\log(1 - v_k) + \sum_{i=1}^{N}\left[\log\Gamma(\sum_{k=1}^{K}\alpha\beta_k) - \sum_{k=1}^{K}\log\Gamma(\alpha\beta_k) + \sum_{k=1}^{K}(\alpha\beta_k - 1)\mathbb{E}_q[\log\pi_{ik}]\right] \tag{3}$$

$$+ \sum_{k=1}^{K}\left[\log\left(\frac{\Gamma(\tau_a + \tau_b)}{\Gamma(\tau_a)\Gamma(\tau_b)}\right) + (\tau_a - 1)\mathbb{E}_q[\log(w_k)] + (\tau_b - 1)\mathbb{E}_q[\log(1 - w_k)]\right] \tag{4}$$

$$- \sum_{ij}^{E}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\phi_{ijk\ell}\log(\phi_{ijk\ell}) \tag{5}$$

$$- \sum_{i=1}^{N}\left[\log\Gamma(\sum_{k=1}^{K}\theta_{ik}) - \sum_{k=1}^{K}\log\Gamma(\theta_{ik}) + \sum_{k=1}^{K}(\theta_{ik} - 1)\mathbb{E}_q[\log\pi_{ik}]\right] \tag{6}$$

$$- \sum_{k=1}^{K}\left[\log\left(\frac{\Gamma(\lambda_{ka} + \lambda_{kb})}{\Gamma(\lambda_{ka})\Gamma(\lambda_{kb})}\right) + (\lambda_{ka} - 1)\mathbb{E}_q[\log(w_k)] + (\lambda_{kb} - 1)\mathbb{E}_q[\log(1 - w_k)]\right] \tag{7}$$

where $\beta_k = v_k\prod_{\ell=1}^{k-1}(1 - v_\ell)$ and $\beta_{K+} = 1 - \sum_{k=1}^{K}\beta_k$. Since we no longer estimate $\phi_{ijk\ell}$ directly, we can show how our ELBO is modified with this optimized inference procedure. In particular, we focus on equations 21, and 24:

$$\sum_{ij}^{E}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\left[\phi_{ijk\ell}(\mathbb{E}_q[\log(\pi_{ik})] + \mathbb{E}_q[\log(\pi_{j\ell})])\right] = \sum_{ij}^{E}\left[\sum_{k=1}^{K}\mathbb{E}_q[\log(\pi_{ik})]\sum_{\ell=1}^{K}\phi_{ijk\ell} + \sum_{\ell=1}^{K}\mathbb{E}_q[\log(\pi_{j\ell})]\sum_{k=1}^{K}\phi_{ijk\ell}\right]$$

$$\sum_{ij}^{E}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\left[\phi_{ijk\ell}\log(\phi_{ijk\ell})\right] = \sum_{ij}^{E}\left[\sum_{k=1}^{K}\mathbb{E}_q[\log(\pi_{ik})]\sum_{\ell=1}^{K}\phi_{ijk\ell} + \sum_{\ell=1}^{K}\mathbb{E}_q[\log(\pi_{j\ell})]\sum_{k=1}^{K}\phi_{ijk\ell}\right.$$
$$\left. + \log f(\epsilon, y_{ij}) - \log f(\epsilon, y_{ij})\sum_{k=1}^{K}\phi_{ijkk} + \sum_{k=1}^{K}\log f(w_k, y_{ij})\phi_{ijkk} - \log(Z_{ij})\right] \tag{8}$$

For an efficient calculation of our ELBO the terms that we need to simplify are $\sum_{k=1}^{K}\mathbb{E}_q[\log(\pi_{ik})]\sum_{\ell=1}^{K}\phi_{ijk\ell}$ and $\sum_{\ell=1}^{K}\mathbb{E}_q[\log(\pi_{j\ell})]\sum_{k=1}^{K}\phi_{ijk\ell}$. From Equation 12, note that

$$\sum_{k=1}^{K}\mathbb{E}_q[\log(\pi_{ik})]\sum_{\ell=1}^{K}\phi_{ijk\ell} = \sum_{k=1}^{K}\tilde{\pi}_{ik}\left[\phi_{ijkk} + \frac{1}{Z_{ij}}\tilde{\pi}_{ik}f(\epsilon, y_{ij})(\tilde{\pi}_j - \tilde{\pi}_{jk})\right]$$

$$\sum_{\ell=1}^{K}\mathbb{E}_q[\log(\pi_{j\ell})]\sum_{k=1}^{K}\phi_{ijk\ell} = \sum_{\ell=1}^{K}\tilde{\pi}_{j\ell}\left[\phi_{ij\ell\ell} + \frac{1}{Z_{ij}}\tilde{\pi}_{j\ell}f(\epsilon, y_{ij})(\tilde{\pi}_i - \tilde{\pi}_{i\ell})\right]$$


1


Note the similarity of this expression with part of the updates in Equation 12. By caching the necessary statistics needed to update $\theta$, we can calculate our ELBO in an efficient manner.

### A.3 Updates for the global stick-breaking weights $\beta$

The global stick breaking weights $\beta$ is not conjugate to node membership weights $\pi$. In order to obtain point estimates for $\beta$ we perform a two-metric constrained optimization using its first order gradients. We can write the objective for $\beta$ w.r.t to our ELBO in the following manner:

$$\mathcal{L}(\beta) = \sum_{k=1}^{K}(\gamma - 1)\log(1 - v_k) - N\sum_{k=1}^{K}\log\Gamma(\alpha\beta_k) + \sum_{k=1}^{K}(\alpha\beta_k - 1)\sum_{i=1}^{N}\mathbb{E}_q[\log\pi_{ik}] \tag{9}$$

Since $\beta_k = v_k\prod_{\ell=1}^{k-1}(1 - v_\ell)$ and $\beta_{K+} = 1 - \sum_{k=1}^{K}\beta_k$, we redefine the prior over $\beta$ as a sum over independently distributed beta variables $v$. We can obtain point estimates for $v$ without having to worry about the constrained optimization task for $\beta$ which is significantly more costly then the two-metric constrained optimization over $v$. We now take the derivatives for $\beta$ with this in mind:

$$\frac{d\mathcal{L}(\beta)}{dv_m} = \frac{-(\gamma-1)}{(1-v_m)} - \alpha N\sum_{k=1}^{K}\frac{d\beta_k}{dv_m}\psi(\alpha\beta_k) + \alpha\sum_{k=1}^{K}\frac{d\beta_k}{dv_m}\sum_{i=1}^{N}\mathbb{E}_q[\log\pi_{ik}] \tag{10}$$

where the derivative $\frac{d\beta_k}{dv_m}$ will change depending on the value of $k$. When $m > k$, then $\frac{d\beta_k}{dv_m} = 0$. When $m = k$, then $\frac{d\beta_k}{dv_m} = \frac{\beta_k}{v_m}$. Finally, when $m < k$, then $\frac{d\beta_k}{dv_m} = \frac{-\beta_k}{(1-v_m)}$.

Our constrained optimization provides us with updates for $v*$ at iteration $t$ which we can then use in our stochastic variational approach by setting $v_k^t = (1 - \rho_t)v_k^{t-1} + \rho_t(v_k^*)$. From this we can determine a new set of values for $\beta^t$ by setting $\beta_k^t = v_k^t\prod_{\ell=1}^{k-1}(1 - v_\ell^t)$.

2

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

## A.4    LittleSis Network Degree-based Visualization
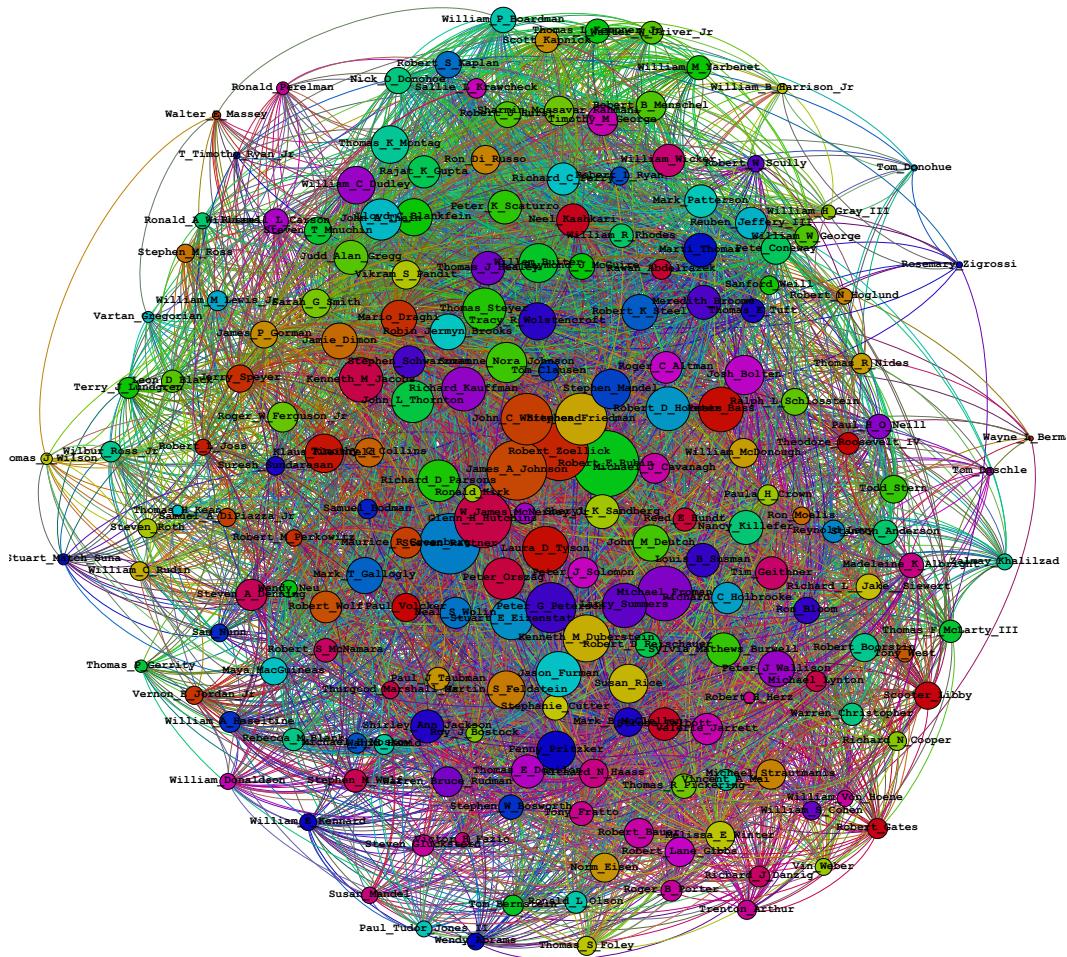
Figure 1: The same raw graph of the top 200 degree nodes is displayed using Gephi and the force atlas layout algorithm. Node sizes were determined by its degree and the raw graph represents a cluttered and un-informative view of its underlying structure. We extracted the original graph from its open source database (http://littlesis.org), which was originally a bipartite graph between individuals and the organizations they were involved in. Other types of relationships such as campaign contributions or shared education can also be extracted, but for this study we focused on whether an individual was a member of that organization. We removed individuals and organizations that appeared only once and to generate an undirected network, we assumed an edge existed between people who held positions within the same organization. The largest connected component was found to contain 18,831 nodes and 626,881 edges which we used as our final graph for analysis.

3