Bayesian Vertex Nomination

Dominic S. Lee[1] and Carey E. Priebe[2]

[1]University of Canterbury, New Zealand; [2]Johns Hopkins University, USA

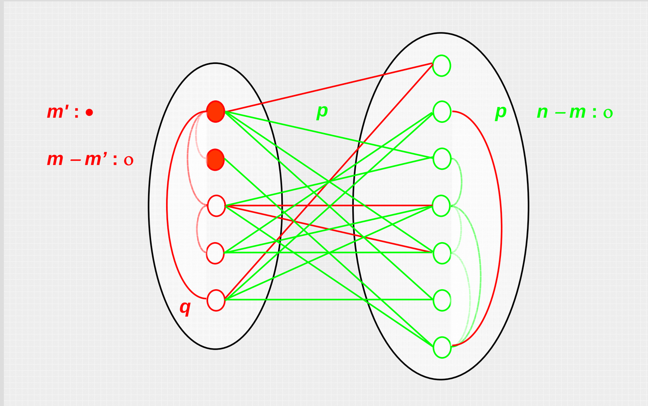
1. Introduction

Suppose we have a community containing a small number of interesting subjects. The identities of these interesting subjects are not fully known; only a few of them are known. The *vertex nomination problem* is to nominate one of the unknown subjects as interesting.

Our approach uses an attributed graph to model the community, with vertices representing subjects, a binary vertex attribute representing whether a subject is interesting, edges representing communications between subjects, and edge attributes representing contents of communications.

Binary edge attributes for the graph shown in the figure. 0 = no edge.

	2	3	4	5	6	7	8	9	10	11	12
1	2	2	0	2	2	0	1	0	1	1	0
2		0	0	0	0	0	0	0	0	0	1
3			2	0	0	0	1	2	2	0	1
4				0	0	1	0	1	1	0	0
5					2	1	0	1	0	1	0
6						0	0	0	0	0	0
7							0	0	0	0	2
8								1	0	0	0
9									1	0	1
10										0	1
11											0



The figure shows a graph with n = 12 vertices, where m' = 2 are known to be interesting (•), $m - m'$ = 3 are interesting but unknown (o), and $n - m$ = 7 are uninteresting and unknown (o). Edge attributes shown are binary (green or red) and all edges and their attributes are assumed to be known. The goal is to pick a red vertex from the vertices whose attributes are unknown.

We formulate a Bayesian model using a *context statistic* (who communicates with who) and a *content statistic* (communication topic) associated with the graph, and assuming that these statistics are independent between vertices and that interesting content is more likely between interesting subjects. A Metropolis-within-Gibbs algorithm is implemented for sampling from the posterior distribution. The nominated vertex is one with the highest posterior probability of being interesting.

2. Models

Notations:

For a vertex v, let

$Y(v)$ = vertex attribute of $v \in \{1 = \text{green}, 2 = \text{red}\}$;
$R(v)$ = context statistic = number of observed red vertices connected to v;
$S(v)$ = content statistic = number of red edges incident to v;
$T(v) = (R(v), S(v))$.

Given two green vertices or two vertices with different color, let

$p_1 = P$(green edge between them);
$p_2 = P$(red edge between them);
$p_0 = P$(no edge between them) $= 1 - p_1 - p_2$.

Given two red vertices, let

$q_1 = P$(green edge between them);
$q_2 = P$(red edge between them);
$q_0 = P$(no edge between them) $= 1 - q_1 - q_2$.

$q = (q_0, q_1, q_2)$ quantifies the frequency and distribution of communication between red vertices, while $p = (p_0, p_1, p_2)$ quantifies these for the rest of the graph (see figure above).

Note that the total number of red vertices is m, of which m' are observed to be red, and $m - m'$ are unobserved or latent. The number of latent green vertices is $n - m$.

Assumptions:

(i) Pairs of red vertices, both observed and latent, communicate with a different frequency from other pairs.

(ii) Distribution of content amongst red vertices is different from the rest of the graph.

(iii) Context and content statistics are independent between vertices.

Specifically, for (i) and (ii), we assume that $p_1 = q_1$ and $p_2 < q_2$.

Models:

Given that v is a green vertex (o), the distribution of $T(v)$ is

$$f_1(T(v) \mid p_1, p_2) = \{\mathrm{Bin}(n-m'-1, p_2) * \mathrm{Bin}(R(v), p_2/(p_1+p_2))\} \cdot \mathrm{Bin}(m', p_1+p_2),$$

where $\mathrm{Bin}(n, p)$ denotes a binomial distribution with parameters n and p, and $g * h$ denotes a discrete convolution between g and h.

Given that v is a latent red vertex (o),

$$f_2(T(v) \mid m, p_1, p_2, q_2) = \{\mathrm{Bin}(n-m, p_2) * \mathrm{Bin}(m-m'-1, q_2) * \mathrm{Bin}(R(v), q_2/(p_1+q_2))\} \cdot \mathrm{Bin}(m', p_1+q_2).$$

Given that v is an observed red vertex (•),

$$f'(T(v) \mid m, p_1, p_2, q_2) = \{\mathrm{Bin}(n-m, p_2) * \mathrm{Bin}(m-m', q_2) * \mathrm{Bin}(R(v), q_2/(p_1+q_2))\} \cdot \mathrm{Bin}(m'-1, p_1+q_2).$$

Let $\mathbf{T'} = (T'(1),\ldots,T'(m'))$ be the statistics for the observed red vertices.
Let $\mathbf{T} = (T(1),\ldots,T(n-m'))$ be the statistics for the latent vertices whose attributes, $\mathbf{Y} = (Y(1),\ldots,Y(n-m'))$, are unknown.

Likelihood function:

$$f(\mathbf{T},\mathbf{T}' \mid \mathbf{Y}, p_1, p_2, q_2) = \prod_{i:Y(i)=1} f_1(T(i) \mid p_1, p_2) \prod_{j:Y(j)=2} f_2(T(j) \mid m, p_1, p_2, q_2) \prod_{k=1}^{m'} f'(T'(k) \mid m, p_1, p_2, q_2),$$

where $m = m' + \sum_{i=1}^{n-m'} I_{\{2\}}(Y(i))$.

Prior distribution:

$$f(\mathbf{Y}, p_1, p_2, q_2 \mid \psi) = f(\mathbf{Y} \mid \psi)\, f(p_1, p_2, q_2),$$

$$f(\mathbf{Y} \mid \psi) = \prod_{i=1}^{n-m'} \mathit{Bernoulli}(\psi) = \psi^{m-m'}(1-\psi)^{n-m},$$

$$f(p_1, p_2, q_2) = f(q_2 \mid p_1, p_2)\, f(p_1, p_2) = \mathit{Uniform}(p_2, 1-p_1)\, \mathit{Dirichlet}(1, 1, 1),$$

$$f(\psi \mid \alpha, \beta) = \mathit{Beta}(\alpha, \beta).$$

Posterior distribution:

$$f(\mathbf{Y}, p_1, p_2, q_2, \psi \mid \mathbf{T}, \mathbf{T}') \propto f(\mathbf{T}, \mathbf{T}' \mid \mathbf{Y}, p_1, p_2, q_2)\, \psi^{m-m'+\alpha-1}(1-\psi)^{n-m+\beta-1} I_{(0,1)}(\psi)$$
$$\cdot (1 - p_1 - p_2)^{-1} I_{(0,1)}(p_1) I_{(0,1-p_1)}(p_2) I_{(p_2,1-p_1)}(q_2).$$

3. Inference

Let $\mathbf{Y}_{-i} = \mathbf{Y} \setminus Y(i)$, and let γ_i be the conditional posterior probability that the latent vertex i is red given the attributes $\mathbf{Y}_{-i}$. Then

$$\frac{1}{\gamma_i(\mathbf{Y}_{-i}, p_1, p_2, q_2, \psi)} = 1 + \frac{f(Y(i)=1, \mathbf{Y}_{-i}, p_1, p_2, q_2, \psi \mid \mathbf{T}, \mathbf{T}')}{f(Y(i)=2, \mathbf{Y}_{-i}, p_1, p_2, q_2, \psi \mid \mathbf{T}, \mathbf{T}')},$$

which allows $\mathbf{Y}$ to be updated component-wise by Gibbs sampling.

ψ can also be updated by Gibbs sampling, since

$$f(\psi \mid \mathbf{T}, \mathbf{T'}, \mathbf{Y}, p_1, p_2, q_2) = \mathit{Beta}(\alpha+m-m', \beta+n-m).$$

p_1, p_2, and q_2, however, must be updated using Metropolis-Hastings sampling with the following proposal distributions derived from the prior distribution:

$$f(p_1 \mid p_2, q_2) = (1-p_1-p_2)^{-1} [\log(1-p_2) - \log(q_2-p_2)]^{-1},\ p_1 \in (0, 1-p_2),$$

$$f(p_2 \mid p_1, q_2) = (1-p_1-p_2)^{-1} [\log(1-p_1) - \log(1-p_1-q_2)]^{-1},\ p_2 \in (0, q_2),$$

$$f(q_2 \mid p_1, p_2) = (1-p_1-p_2)^{-1},\ q_2 \in (p_2, 1-p_1).$$

Metropolis-within-Gibbs sampler:

Let $(\mathbf{Y}^{(h)}, p_1^{(h)}, p_2^{(h)}, q_2^{(h)}, \psi^{(h)})$ denote the state at iteration h.

Gibbs step:

For $i = 1, \ldots, n-m'$:

Compute $\gamma_i(Y^{(h)}(1),\ldots,Y^{(h)}(i-1),Y^{(h-1)}(i+1),Y^{(h-1)}(n-m'),p_1^{(h-1)},p_2^{(h-1)},q_2^{(h-1)},\psi^{(h-1)})$,

Set $Y^{(h)}(i)$ = 1 or 2 with probability $1-\gamma_i$ and γ_i respectively.

Compute $m^{(h)} = m' + \sum I_{\{2\}}(Y^{(h)}(i))$.

Generate $\psi^{(h)} \sim \mathit{Beta}(\alpha+m^{(h)}-m', \beta+n-m^{(h)})$.

Metropolis-Hastings step:

Generate $p_1^* \sim f(p_1 \mid p_2^{(h-1)}, q_2^{(h-1)})$.

Compute $\pi(p_1) = \min\left\{1, \dfrac{f(\mathbf{T},\mathbf{T}' \mid \mathbf{Y}^{(h)}, p_1^*, p_2^{(h-1)}, q_2^{(h-1)})}{f(\mathbf{T},\mathbf{T}' \mid \mathbf{Y}^{(h)}, p_1^{(h-1)}, p_2^{(h-1)}, q_2^{(h-1)})}\right\}$.

Set $p_1^{(h)} = p_1^*$ or $p_1^{(h-1)}$ with probability $\pi(p_1)$ and $1-\pi(p_1)$ respectively.

Generate $p_2^* \sim f(p_2 \mid p_1^{(h)}, q_2^{(h-1)})$.

Compute $\pi(p_2) = \min\left\{1, \dfrac{f(\mathbf{T},\mathbf{T}' \mid \mathbf{Y}^{(h)}, p_1^{(h)}, p_2^*, q_2^{(h-1)})}{f(\mathbf{T},\mathbf{T}' \mid \mathbf{Y}^{(h)}, p_1^{(h)}, p_2^{(h-1)}, q_2^{(h-1)})}\right\}$

Set $p_2^{(h)} = p_2^*$ or $p_2^{(h-1)}$ with probability $\pi(p_2)$ and $1-\pi(p_2)$ respectively.

Generate $q_2^* \sim f(q_2 \mid p_1^{(h)}, p_2^{(h)})$.

Compute $\pi(q_2) = \min\left\{1, \dfrac{f(\mathbf{T},\mathbf{T}' \mid \mathbf{Y}^{(h)}, p_1^{(h)}, p_2^{(h)}, q_2^*)}{f(\mathbf{T},\mathbf{T}' \mid \mathbf{Y}^{(h)}, p_1^{(h)}, p_2^{(h)}, q_2^{(h-1)})}\right\}$

Set $q_2^{(h)} = q_2^*$ or $q_2^{(h-1)}$ with probability $\pi(q_2)$ and $1-\pi(q_2)$ respectively.

Hyperprior distribution:

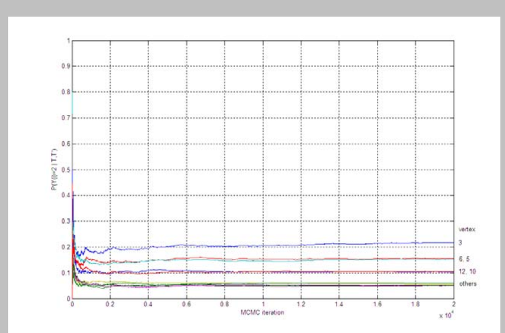
Since our goal is to nominate a *single* vertex, the beta hyperprior distribution for ψ is chosen to induce sparsity in the potential nominees. One way to achieve this is a beta density with mode at $1/(n-m')$; a convenient choice being $\alpha = 2$ and $\beta = n-m'$.

4. Simulation Results

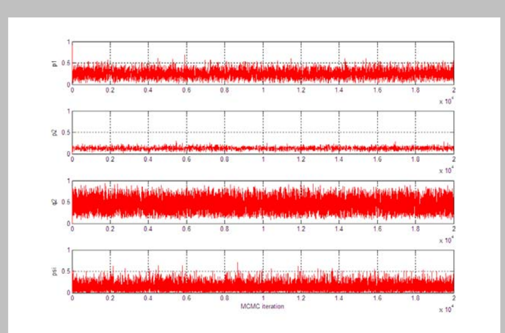
Experiment 1: n = 12, m = 5, m' = 2, p_1 = 0.25, p_2 = 0.15, q_2 = 0.25

Results for the graph shown above:

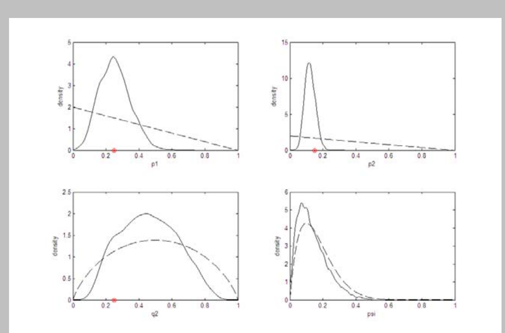
Trace plots of the moving average estimates of the marginal posterior probabilities that each of the unlabelled vertices is red. The top-ranking vertex is vertex 3, which is a latent red vertex and so we have a correct nomination in this case.



Trace plots of nuisance parameters, p_1, p_2, q_2, and hyperparameter, ψ.



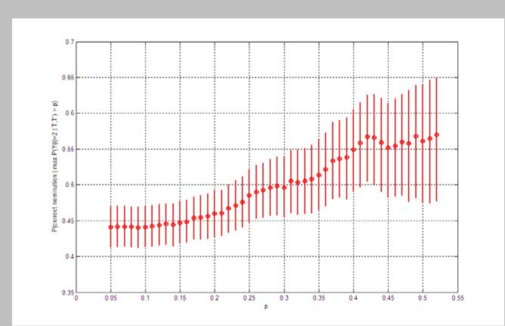
Marginal prior densities (dashed curves) and posterior densities (solid curves) for p_1, p_2, q_2 and ψ. Red points on the x-axis indicate the true parameter values. Notice the concentration of the posterior densities near the true values for p_1 and p_2 but less so for q_2 because of the smaller number of red vertices.



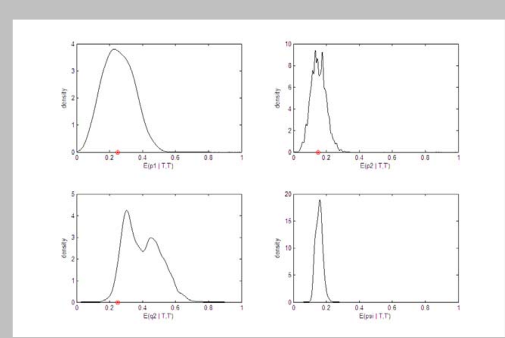
Results from 1000 graphs:

Probability of correct nomination ≈ 0.44,
95% BCA bootstrap confidence interval = (0.41, 0.47),
Probability of correct nomination purely by chance is 3/10 = 0.3,
Odds ratio for correct nomination relative to chance ≈ 1.8.

Estimates of the conditional probability of correct nomination given that the marginal posterior probability that the nominated vertex is red exceeds p, for values of p on the x-axis. Notice that an increasing trend is evident.



Kernel densities fitted to posterior means of p_1, p_2, q_2 and ψ from the 1000 graphs. Observe the concentration of probability mass around the true values of the nuisance parameters, indicated by red points on the x-axis.



Experiment 2: n = 184, p_1 = 0.2, p_2 = 0.2, q_2 = 0.4

Coppersmith & Priebe (2012) defined a linear fusion statistic for vertex v, combining its context and content statistics, as

$$\tau_\lambda(v) = (1-\lambda)R(v) + \lambda S(v),$$

where $\lambda \in [0, 1]$ is a fusion parameter that determined the relative weight of context and content information. For a given value of λ, the nominated vertex was a latent vertex with the largest value of τ_λ.

The table below compares our method (BVN) with that of Coppersmith & Priebe (C&P), in terms of the probability of correct nomination for selected values of m and m'. 1000 graphs were used for each pair of (m, m') values. Observe that when m' is small relative to m, the two methods have the same performance. However, as m' increases relative to m, BVN performs increasingly better than C&P, and has a higher rate of improvement when m is larger.

m = 8

	m' = 2	m' = 4	m' = 6
BVN	0.09 (0.08, 0.10)†	0.12 (0.10, 0.13)	0.09 (0.08, 0.11)
C&P††	0.09	0.11	0.06
OR†††	1	1.10	1.55

m = 32

	m' = 8	m' = 16	m' = 24
BVN	0.83 (0.81, 0.85)	0.90 (0.88, 0.92)	0.87 (0.85, 0.89)
C&P	0.83	0.86	0.78
OR	1	1.47	1.89

† 95% BCA bootstrap confidence interval.
†† Optimal performance with optimal fusion parameter.
††† Odds ratio for correct nomination by BVN relative to C&P.

5. Application Results

The Enron email corpus, available at http://www.enron-mail.com/, consists of email communications amongst Enron employees and their associates. Some of them were allegedly committing fraud and their fraudulent activity was captured in some emails along with many innocuous ones. Priebe, et al. (2005) derived a processed version of a subset of the email data, over a period of 189 weeks from 1998 to 2002. This yielded 1 graph per week, each containing the same 184 email users forming the vertices of the graph; 10 of these users have been found to have committed fraud. Berry, et al. (2007) indexed the contents of a subset of the email corpus into 32 topics. These same topics were adopted by Coppersmith & Priebe (2012), who introduced a mapping from the topics to a binary edge attribute denoting content perceived as innocuous or fraudulent.

We used one of the graphs derived by Priebe, et al. (2005), together with the binary edge attributes from Coppersmith & Priebe (2012), for the experiments described here.

Experiment 1:
5 of the 10 fraudsters were treated as known and the others as unknown, to see whether one of the unknown fraudsters will be correctly nominated. Thus, n = 184, m = 10 and m' = 5. The probability of correct nomination was estimated from all 252 (10 choose 5) combinations of 5 known fraudsters taken from the 10 fraudsters. For each combination, 1000 MCMC iterations were used for estimation after a burn-in of 1000 iterations.

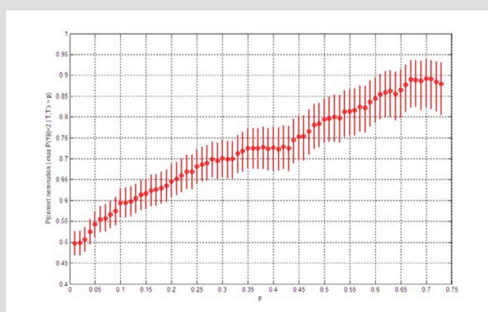
Note that, in this case, the probability of correct nomination purely by chance is 5/179 ≈ 0.03.

Results:
Probability of correct nomination ≈ 0.10,
95% BCA bootstrap confidence interval = (0.09, 0.11),
Odds ratio for correct nomination relative to chance ≈ 3.6.
Sample means of the posterior means from the 252 combinations are p_1 = 0.0168, p_2 = 0.0111 and q_2 = 0.1298.

Experiment 2:
Estimates of p_1, p_2 and q_2 from Experiment 1 were treated as true values in a Monte Carlo simulation involving n = 184, m = 10 and m' = 5.

Results from 1000 graphs:
Probability of correct nomination ≈ 0.50,
95% BCA bootstrap confidence interval = (0.47, 0.53),
Odds ratio for correct nomination relative to chance ≈ 32.3.

Once again, we have an increasing trend in the conditional probability of correct nomination given that the marginal posterior probability that the nominated vertex is red exceeds p. This trend is even more pronounced here than before.



6. Conclusion

The Bayesian model

(i) performs significantly better than chance;

(ii) gives a probability of correct nomination that increases with increasing posterior probability that the nominated vertex is red;

(iii) matches or performs better than the method in Coppersmith & Priebe (2012).

A full paper is available from arXiv:1205.5082

References

Berry, M.W., Browne, M. and Signer, B. (2007). 2001 Topic annotated Enron email data set. Linguistic Data Consortium. Philadelphia.

Coppersmith, G.A. and Priebe, C.E. (2012). Vertex nomination via content and context. arXiv:1201.4118v1, 19 January 2012.

Priebe, C. E., Conroy, J. M., Marchette, D. J. and Park, Y. (2005). Scan statistics on Enron graphs. Computational and Mathematical Organization Theory 11, 229-247.

Acknowledgements

Thanks to Youngser Park for assistance with the Enron data, and to Celine Cattoen-Gilbert for suggestions about computations.