

Vertex Nomination Schemes for Membership Prediction

D.E. Fishkind, V. Lyzinski, H. Pao, L. Chen, C.E. Priebe

Department of Applied Mathematics and Statistics, Johns Hopkins University

August 8, 2014

Abstract

Suppose that a graph is realized from a stochastic block model where one of the blocks is of interest, but many or all of the graph vertices' block labels are unobserved. The task is to order the vertices with unobserved block labels into a “nomination list” such that, with high probability, vertices from the interesting block are concentrated near the list's beginning. We propose several vertex nomination schemes which include the utilization of recent graph matching and spectral partitioning machinery.

Our basic—but principled—setting and development yields a best nomination scheme (which is a Bayes-Optimal analogue), and a likelihood maximization nomination scheme that is practical to implement when there are a thousand vertices, and which is empirically near-optimal when the number of vertices is small enough to allow the comparison. We then illustrate the robustness of likelihood maximization to the modeling challenges inherent in real data, using examples which include the Enron Graph, a worm brain connectome, and a political blog network.

1 Article overview

In a stochastic block model, the vertices are partitioned into blocks, and existence/nonexistence of an edge between any pair of vertices is an independent Bernoulli trial, with the Bernoulli parameter being a function of the block memberships of the pair of vertices. We are concerned here with a graph realized from a stochastic block model such that many or all of the vertices' block labels are hidden (unobserved). Suppose that one particular block is of interest, and the task is to order

the vertices with hidden block label into a “nomination list” with the goal of having vertices of the interesting block concentrated near the beginning of the list. Forming such a nomination list can be assisted by any available knowledge about the underlying model parameters, as well as by utilizing knowledge of block membership for any of the vertices for which such block labels are observed. A vertex nomination scheme is a function that, to each such possible observed graph, assigns an associated nomination list. In this paper we present, analyze, and illustrate the effectiveness of several vertex nomination schemes. Some of these vertex nomination schemes utilize graph matching and spectral partitioning machinery. See [6] and [13] for recent work on vertex nomination, as well as a survey of closely related problems.

One illustrative example of vertex nomination with real data is the political blogosphere data example, treated in greater detail later in Section 7.3. The graph vertices are a set of political blogs, and graph adjacency between two blogs is defined as one blog citing the other. This graph can be modeled (very roughly) as being an instantiation of a stochastic block model with two blocks; namely, the liberal blogs and the conservative blogs. Suppose you work for a political action committee, and want to read as many liberal blogs as possible over a limited amount of time, in order to write a report about the liberal position on some issue of interest in current events. Every one of the blogs is either liberal or conservative, but suppose that you know a-priori the political philosophy of only a very few of the blogs. However, you do observe the entire graph. Since your blog-reading time is limited, you would like to use this partial knowledge to prioritize the rest of the blogs (whose political philosophy is not known a-priori) into a reading list (i.e. nomination list) in such a manner that there will be an abundance of liberal blogs at the beginning of your reading list. In particular, the nomination task here is a task which is not simply classification—it is prioritization.

In Section 2 we formally and carefully define the setting and the concept of a vertex nomination scheme. Although prioritization is a ubiquitous need that can be treated in an ad hoc fashion specific to individual applications, we here formally set the problem in the stochastic block model setting, which has gained so much popularity in recent literature (e.g., see [2, 4, 20]) and is a useful model for real data. This formal setting will be useful for principled development of techniques that have both solid theoretical foundations are also robust to the modeling challenges inherent in real data.

In Section 3 we introduce the canonical vertex nomination scheme. It is analogous to the Bayes classifier in the setting of classification. Indeed, we prove in Proposition 1 that the canonical vertex nomination scheme is at least as effective as every other vertex nomination scheme, and thus serves the valuable role of a “gold standard” with which to gauge the success of other vertex nomination

schemes. However it is computationally practical to implement only when there are on the order of a very few tens of vertices.

In Section 4 we introduce the likelihood maximization vertex nomination scheme, which fundamentally utilizes graph matching machinery. The graph matching problem is to find a bijection between the vertex sets of two graphs that minimizes the number of induced adjacency disagreements; there is a vast literature dedicated to this problem, e.g. see the 2004 article *Thirty Years of Graph Matching in Pattern Recognition* [5] for an excellent survey. Although graph matching is intractable in theory, there have been recent advances in approximate graph matching algorithms that are both tractable and effective; for example see [16], [25], and [26]. In particular, the very recent SGM algorithm of [16] has been shown in [15] to be theoretically and practically superior to convex relaxation approaches. Using the SGM algorithm of [16] for approximate graph matching, the likelihood maximization vertex nomination scheme is practical to implement for on the order of 1000 vertices. In Sections 7.1, 7.2, 7.3, and 7.4 we illustrate the robustness of the likelihood maximization vertex nomination scheme to the model mis-specifications inherent in real data. On the other hand, when there are few enough vertices so that canonical is computable, we find in Section 6 that likelihood maximization performs nearly as well as the canonical “gold standard.”

In Section 5 we introduce the spectral partitioning vertex nomination scheme; it is practical to implement for tens of thousands of vertices or more. Based on the results in [24] and [8], then followed up in [18], the spectral partitioning vertex nomination scheme nominates perfectly as the number of vertices goes to infinity, under mild conditions.

In Section 6 we perform illustrative simulations at three different scales; that is, a “small scale” experiment with ten ambiguous vertices, a “medium scale” experiment with 500 ambiguous vertices, and a “large scale” experiment with 10000 ambiguous vertices. With respect to nomination effectiveness and practicality of implementation, the canonical vertex nomination scheme dominates at the small scale, the likelihood maximization scheme dominates at the medium scale, and the spectral partitioning scheme dominates at the large scale.

In Section 7.1 we illustrate our vertex nomination schemes on the “Enron Graph,” a graph with email addresses of former employees of the failed Enron Corporation as vertices, and edges indicating email contact between the associated vertices over a time interval. Our vertex nomination schemes are used to nominate higher-echelon former Enron employees. Then, in Sections 7.2, 7.3, and 7.4 we illustrate on examples with a worm-brain connectome (to nominate motor neurons), a blog network (to nominate political affiliation), and a movie network (to nominate comedies).

We conclude the paper with an extended discussion in Section 8, and we describe how this work paves the way for the next generation of vertex nomination schemes.

2 Vertex nomination schemes; setting and definition

In this article we assume for simplicity that graphs are simple (edges are not directed, no parallel edges, and no single-edge loops), but much of what we do is generalizable.

We begin by describing the stochastic block distribution $\text{SB}(K, m, n, b, \Lambda)$, which will be our random graph setting; its parameters are a positive integer K (the number of *blocks*), a nonnegative integer m (the number of *seeds*), a positive integer n (the number of *ambiguous vertices*), an arbitrary but fixed function $b : \{1, 2, \dots, m+n\} \rightarrow \{1, 2, \dots, K\}$ (the *block membership function*), and a symmetric matrix $\Lambda \in [0, 1]^{K \times K}$ (the *adjacency probabilities*). A random graph with distribution $\text{SB}(K, m, n, b, \Lambda)$ has the vertex set $W := \{1, 2, \dots, m+n\}$ and, for each unordered pair of distinct vertices $\{w, w'\} \in \binom{W}{2}$, w is adjacent to w' ($w \sim w'$) according to an independent Bernoulli trial with parameter $\Lambda_{b(w), b(w')}$.

The vertex set W is partitioned into two sets, the set $U := \{1, 2, \dots, m\}$ (the *seeds*) and the set $V := \{m+1, m+2, \dots, m+n\}$ (the *ambiguous vertices*). For each $i = 1, 2, \dots, K$, define $m_i := |\{u \in U : b(u) = i\}|$ and $n_i := |\{v \in V : b(v) = i\}|$. The function b is only partially observed; its values are known on U , but not on V . In other words, the block memberships of the seeds are known, and the block memberships of the ambiguous vertices are unknown, but we will assume for simplicity that Λ is known, and that n_1, n_2, \dots, n_K are known. Given a random graph from $\text{SB}(K, m, n, b, \Lambda)$, the most general inferential task would be to estimate b on W , but we will fine-tune this task very soon. (Note that if Λ and n_1, n_2, \dots, n_K were not known then, if there are enough seeds, Λ could be approximated from edge densities of subgraphs induced by various subsets of the seeds and, in addition, the values of n_1, n_2, \dots, n_K might be approximated if it just so happens to be known that they are roughly proportional to the respective values of m_1, m_2, \dots, m_K . Of course, m_1, m_2, \dots, m_K are known by virtue of the fact that b is known on U .)

Define Ξ to be the set of bijective functions from W to W that fix the elements of U ; of course, $|\Xi| = n!$. Any two graphs G and H on vertex set W are called *equivalent* if G is isomorphic to H under some function $\xi \in \Xi$; if G is also asymmetric (i.e. its automorphism group is trivial) then such a ξ is unique to G, H , denote it $\xi_{G,H}$. For any graph G on vertex set W , the equivalence class of equivalent-to- G graphs on vertex set W will be denoted $\langle G \rangle$; in particular, $\langle G \rangle$ is an event. The set of all such equivalence classes is denoted Θ ; the events in Θ partition the sample space.

A *vertex nomination scheme* Φ is a mapping that, to each asymmetric graph G with vertex set W , associates a linear ordering of the vertices in V —called the *nomination order*, and denoted as a list $(\Phi_G(1), \Phi_G(2), \dots, \Phi_G(n))$ —such that for every H equivalent to G it holds that $(\xi_{G,H}(\Phi_G(1)), \xi_{G,H}(\Phi_G(2)), \dots, \xi_{G,H}(\Phi_G(n))) = (\Phi_H(1), \Phi_H(2), \dots, \Phi_H(n))$. In other words, and

described somewhat informally, if each equivalence class of graphs is viewed as a (single) graph whose vertex set is comprised of labeled vertices U and unlabeled vertices V , then to each equivalence class (i.e. partially-vertex-labeled graph) Φ associates a list of unlabeled vertices of V .

Note that the fraction of all graphs on vertex set W which are symmetric goes very quickly to zero as $|W|$ goes to infinity [7, 21]. Although symmetric graphs are thus negligibly-many, it is helpful for notation to extend the domain of Φ to include symmetric graphs, and this can be done in many different ways. For simplicity of analysis we will simply say for now that, to every symmetric graph G on the vertex set W , the associated nomination list is declared to be $(m + 1, m + 2, \dots, m + n)$ (and we do not require the nomination list in this case to meet the property mentioned above).

In this article, we assume that only membership in the first block is of interest; the specific task we are concerned with is to find vertex nomination schemes under which there will be, with high probability, an abundance of members of the first block that are near the beginning of the nomination list. As an illustrative example related to the Enron Graph example in Section 7.1, consider a corporation with $m + n = m_1 + m_2 + n_1 + n_2$ employees, of which $m_1 + n_1$ are involved in fraud and $m_2 + n_2$ are not involved in fraud. The probability of communication between fraudsters is fixed, as is the probability of communication between nonfraudsters, as is the probability of communication between any fraudster and any nonfraudster. Of the $m_1 + n_1$ fraudsters, m_1 have been identified as fraudsters and, among the $m_2 + n_2$ nonfraudsters, m_2 have been identified as nonfraudsters. Based on observing all of the employee communications (together with knowledge of the identities of m_1 fraudsters and m_2 nonfraudsters), we wish to draw up a nomination list of the $n_1 + n_2$ ambiguous employees so that there are many fraudsters early in the list.

The effectiveness of a vertex nomination scheme Φ is quantified in the following manner. For any graph G with vertex set W , and for any integer j such that $1 \leq j \leq n$, the *precision at depth j* of Φ for G is defined to be $\frac{|\{1 \leq i \leq j : b(\Phi_G(i))=1\}|}{j}$; for the corporate illustration, this represents the fraction of the first j employees on the nomination list that are actual fraudsters in truth. The *average precision* of Φ for G is defined to be $\frac{1}{n_1} \sum_{j=1}^{n_1} \frac{|\{1 \leq i \leq j : b(\Phi_G(i))=1\}|}{j}$; it has a value between 0 (per the corporate example, if none of the first n_1 nominated employees are fraudsters) and 1 (if all of the first n_1 nominated employees are fraudsters). Note that the average precision of Φ for G is equal to $\sum_{i=1}^{n_1} \left(\frac{1}{n_1} \sum_{j=i}^{n_1} \frac{1}{j} \right) \delta_{b(\Phi_G(i))=1}$, where δ is the usual indicator function. In particular, the average precision of Φ for G is a convex combination of the indicators $\delta_{b(\Phi_G(i))=1}$, with more weight in this convex combination for indicators associated with lower values of i . The *mean average precision* of the vertex nomination scheme Φ is the expected value of the average precision for a random graph G distributed $\text{SB}(K, m, n, b, \Lambda)$. The closer that this number is to 1, the

more effective a vertex nomination scheme Φ is deemed. Note that a “chance” vertex nomination scheme would have the value $\frac{n_1}{n}$ as its mean average precision.

We point out that our definition of average precision is slightly different than a definition commonly used in the information retrieval community; our definition is a pure average precision, whereas the other definition is actually an integral of the precision over recall.

3 The canonical vertex nomination scheme

In this section we define the canonical vertex nomination scheme, which is analogous to the Bayes classifier in the Bayes classifier’s setting of classification. Indeed, we prove in Proposition 1 that the mean average precision of the canonical vertex nomination scheme is greater than or equal to the mean average precision of every other vertex nomination scheme. Unfortunately, because of its computational intractability (a visibly exponential runtime as the number of vertices increases) the canonical vertex nomination scheme is only practical to implement for up to a few tens of vertices. Nonetheless, because of Proposition 1, the canonical vertex nomination scheme serves as a valuable “gold standard” to evaluate the performance of other more computationally tractable vertex nomination schemes. (This is analogous to the role of the Bayes classifier in the classification setting.) Our ongoing research seeks to approximate the canonical vertex nomination scheme in a scalable fashion.

3.1 Definition of the scheme

Consider the random graph G distributed $\text{SB}(K, m, n, b, \Lambda)$. When G is asymmetric then, for any $v \in V$, the conditional probability

$$\mathbb{P}\left[\{H \in \langle G \rangle : b(\xi_{G,H}(v)) = 1\} \middle| \langle G \rangle\right] \quad (1)$$

may be described as the probability, given the event that we observe a graph equivalent to G , that the vertex corresponding to v would be in the first block. The *canonical vertex nomination scheme*, which we denote as Φ^C , orders the vertices of V as $\Phi_G^C(1), \Phi_G^C(2), \dots, \Phi_G^C(n)$ in decreasing order of this conditional probability; that is, we define Φ^C so that, for all $i = 1, 2, \dots, n - 1$,

$$\mathbb{P}\left[\{H \in \langle G \rangle : b(\xi_{G,H}(\Phi_G^C(i))) = 1\} \middle| \langle G \rangle\right] \geq \mathbb{P}\left[\{H \in \langle G \rangle : b(\xi_{G,H}(\Phi_G^C(i+1))) = 1\} \middle| \langle G \rangle\right]. \quad (2)$$

To more easily compute the conditional probability in Equation 1, let $\binom{V}{n_1, n_2, \dots, n_K}$ denote the collection of all the $\binom{n}{n_1, n_2, \dots, n_K}$ partitions of the elements of V into subsets called V_1, V_2, \dots, V_K with

respective cardinalities n_1, n_2, \dots, n_K . Given any such partition $(V_1, V_2, \dots, V_K) \in \binom{V}{n_1, n_2, \dots, n_K}$, let us create the following notation. For any $k = 1, 2, \dots, K$ and $\ell = k+1, k+2, \dots, K$, let $e_{k,\ell}$ denote the number of edges in G with one endpoint in $V_k \cup \{u \in U : b(u) = k\}$ and the other endpoint in $V_\ell \cup \{u \in U : b(u) = \ell\}$, define $c_{k,\ell} := (m_k + n_k)(m_\ell + n_\ell) - e_{k,\ell}$. Let $e_{k,k}$ denote the number of edges in G with both endpoints in $V_k \cup \{u \in U : b(u) = k\}$, and define $c_{k,k} := \binom{m_k + n_k}{2} - e_{k,k}$. Then, in the stochastic block model, the conditional probability in Equation (1) can be computed as

$$\frac{\sum_{(V_1, V_2, \dots, V_K) \in \binom{V}{n_1, n_2, \dots, n_K} \text{ such that } v \in V_1} \prod_{k=1}^K \prod_{\ell=k}^K (\Lambda_{k,\ell})^{e_{k,\ell}} (1 - \Lambda_{k,\ell})^{c_{k,\ell}}}{\sum_{(V_1, V_2, \dots, V_K) \in \binom{V}{n_1, n_2, \dots, n_K}} \prod_{k=1}^K \prod_{\ell=k}^K (\Lambda_{k,\ell})^{e_{k,\ell}} (1 - \Lambda_{k,\ell})^{c_{k,\ell}}}. \quad (3)$$

Although we are not able to evaluate the probability of G since the block membership function b is not fully observed, nonetheless the conditional probabilities in Equation (1) can be indeed be evaluated via Equation (3) by just knowing the values of the parameters n_1, n_2, \dots, n_K and Λ .

3.2 Optimality of the canonical vertex nomination scheme

Theorem 1. *For any vertex nomination scheme Φ , the mean average precision of the canonical vertex nomination scheme Φ^C is greater then or equal to the mean average precision of Φ .*

Proof: For each $i = 1, 2, \dots, n_1$, define $\alpha_i := \frac{1}{n_1} \sum_{j=i}^{n_1} \frac{1}{j}$ and, for each $i = n_1 + 1, n_1 + 2, \dots, n$, define $\alpha_i := 0$. The sequence $\alpha_1, \alpha_2, \dots, \alpha_n$ is clearly a nonnegative, nonincreasing sequence. Note that if a_1, a_2, \dots, a_n is any (other) nonincreasing, nonnegative sequence of real numbers, and a'_1, a'_2, \dots, a'_n is any permutation of the sequence a_1, a_2, \dots, a_n , then

$$\sum_{i=1}^n \alpha_i a'_i \leq \sum_{i=1}^n \alpha_i a_i. \quad (4)$$

Indeed, this is easily verified by first considering particular sequences a_1, a_2, \dots, a_n of the form $1, 1, \dots, 1, 0, \dots, 0, 0$ [i.e., j consecutive 1's followed by $n - j$ consecutive 0's, for different values of $j = 1, 2, \dots, n$] and then noting that the nonnegative combinations of such particular sequences indeed comprise all nonincreasing, nonnegative sequences with n entries.

Consider random graph G distributed $\text{SB}(K, m, n, b, \Lambda)$. Recall that Θ denotes the set of equivalence classes of graphs on the vertex set W .

Expanding the mean average precisions of Φ , then bounding and simplifying, yields

$$\begin{aligned}
\mathbb{E}\left(\sum_{i=1}^n \alpha_i \delta_{b(\Phi_G(i))=1}\right) &= \sum_{i=1}^n \alpha_i \mathbb{P}\left(b(\Phi_G(i)) = 1\right) \\
&= \sum_{i=1}^n \alpha_i \left(\sum_{\mathcal{G} \in \Theta} \mathbb{P}(\mathcal{G}) \mathbb{P}\left(b(\Phi_G(i)) = 1 \mid \mathcal{G}\right) \right) \\
&= \sum_{\mathcal{G} \in \Theta} \mathbb{P}(\mathcal{G}) \left(\sum_{i=1}^n \alpha_i \mathbb{P}\left(b(\Phi_G(i)) = 1 \mid \mathcal{G}\right) \right) \\
&\leq \sum_{\mathcal{G} \in \Theta} \mathbb{P}(\mathcal{G}) \left(\sum_{i=1}^n \alpha_i \mathbb{P}\left(b(\Phi_G^{\mathcal{C}}(i)) = 1 \mid \mathcal{G}\right) \right) \\
&= \sum_{i=1}^n \alpha_i \mathbb{P}\left(b(\Phi_G^{\mathcal{C}}(i)) = 1\right) = \mathbb{E}\left(\sum_{i=1}^n \alpha_i \delta_{b(\Phi_G^{\mathcal{C}}(i))=1}\right),
\end{aligned} \tag{5}$$

where the inequality in Equation (5) follows from Equations (4) and (2), (and from our assumption that all nomination schemes agree when G is symmetric). The desired result is shown. \square

4 Likelihood maximization vertex nomination scheme

In this section we define the likelihood maximization vertex nomination scheme. It will be practical to implement even when there are on the order of a thousand vertices. We will see in Section 6 that it is a very effective vertex nomination scheme, when compared to the canonical vertex nomination scheme “gold standard” on graphs small enough to make the comparison. In Sections 7.1, 7.2, 7.3, and 7.4 we will see that likelihood maximization appears to be nicely robust to the modeling challenges inherent in real data.

4.1 Definition of the scheme

Suppose the random graph G is distributed $\text{SB}(K, m, n, b, \Lambda)$. There are two stages in defining—and computing—the likelihood maximization vertex nomination scheme.

The first stage is concerned with estimating the block assignment function b . Let \mathfrak{B} denote the set of functions $\mathbf{b} : W \rightarrow \{1, 2, \dots, K\}$ such that \mathbf{b} agrees with b on U , and such that it also holds, for all $i = 1, 2, \dots, K$, that $|\{v \in V : \mathbf{b}(v) = i\}| = n_i$. For any $\mathbf{b} \in \mathfrak{B}$, and for all $k = 1, 2, \dots, K$ and $\ell = k + 1, k + 2, \dots, K$, let $e_{k,\ell}(\mathbf{b})$ denote the number of edges in G with one endpoint in $\{w \in W : \mathbf{b}(w) = k\}$ and the other endpoint in $\{w \in W : \mathbf{b}(w) = \ell\}$, and also denote $c_{k,\ell}(\mathbf{b}) := (m_k + n_k)(m_\ell + n_\ell) - e_{k,\ell}(\mathbf{b})$. For all $k = 1, 2, \dots, K$, let $e_{k,k}(\mathbf{b})$ denote the number of edges

in G with both endpoints in $\{w \in W : \mathbf{b}(w) = k\}$, and also denote $c_{k,k}(\mathbf{b}) := \binom{m_k + n_k}{2} - e_{k,k}(\mathbf{b})$. In the $\text{SB}(K, m, n, b, \Lambda)$ distribution, if b had been replaced with $\mathbf{b} \in \mathfrak{B}$, then the probability of realizing the graph G would have been

$$p(\mathbf{b}, G) := \prod_{k=1}^K \prod_{\ell=k}^K (\Lambda_{k,\ell})^{e_{k,\ell}(\mathbf{b})} (1 - \Lambda_{k,\ell})^{c_{k,\ell}(\mathbf{b})}. \quad (6)$$

Define \hat{b} , the *maximum likelihood estimator of b* , to be the member of \mathfrak{B} such that the probability of G is maximized. In other words, (then taking logarithms and ignoring additive constants)

$$\begin{aligned} \hat{b} := \arg \max_{\mathbf{b} \in \mathfrak{B}} p(\mathbf{b}, G) &= \arg \max_{\mathbf{b} \in \mathfrak{B}} \sum_{k=1}^K \sum_{\ell=k}^K e_{k,\ell}(\mathbf{b}) \log \left(\frac{\Lambda_{k,\ell}}{1 - \Lambda_{k,\ell}} \right) \\ &= \arg \max_{\mathbf{b} \in \mathfrak{B}} \sum_{\{w, w'\} \in \binom{W}{2}} \delta_{w \sim_G w'} \log \left(\frac{\Lambda_{\mathbf{b}(w), \mathbf{b}(w')}}{1 - \Lambda_{\mathbf{b}(w), \mathbf{b}(w')}} \right). \end{aligned} \quad (7)$$

The optimization problem in Equation (7) is an example of seeded graph matching, and we can efficiently and effectively approximate its solution. The details of this are deferred to the next section, Section 4.2, and we now continue on to the second stage of defining and computing the likelihood maximization vertex nomination scheme, assuming that we have computed \hat{b} .

For any $v, v' \in V$ such that $\hat{b}(v) = 1$ and $\hat{b}(v') \neq 1$, define $\hat{b}_{v \leftrightarrow v'} \in \mathfrak{B}$ such that $\hat{b}_{v \leftrightarrow v'}$ agrees with \hat{b} for all $w \in W$ except that $\hat{b}_{v \leftrightarrow v'}(v') = 1$ and $\hat{b}_{v \leftrightarrow v'}(v) = \hat{b}(v')$. For any $v, v' \in V$ such that $\hat{b}(v) = 1$ and $\hat{b}(v') \neq 1$, we can interpret a low/high value of the quantity $\frac{p(\hat{b}_{v \leftrightarrow v'}, G)}{p(\hat{b}, G)}$ as a measure of our conviction/lack-of-conviction that \hat{b} should be used to estimate b , as opposed to estimating b with specifically $\hat{b}_{v \leftrightarrow v'}$. In this spirit, for all $v \in V$ such that $\hat{b}(v) = 1$, a low/high value of the geometric mean

$$\left(\prod_{v' \in V: \hat{b}(v') \neq 1} \frac{p(\hat{b}_{v \leftrightarrow v'}, G)}{p(\hat{b}, G)} \right)^{\frac{1}{n - n_1}} \quad (8)$$

can be interpreted as a measure (for the purpose of ordering) of our conviction/lack-of-conviction in our estimation that $b(v)$ is 1. Also, for all $v' \in V$ such that $\hat{b}(v') \neq 1$, a low/high value of the geometric mean

$$\left(\prod_{v \in V: \hat{b}(v) = 1} \frac{p(\hat{b}_{v \leftrightarrow v'}, G)}{p(\hat{b}, G)} \right)^{\frac{1}{n_1}} \quad (9)$$

can be interpreted as a measure (just for the purpose of ordering) of our conviction/lack-of-conviction in our estimation that $b(v')$ is not 1.

We now define the *likelihood maximization vertex nomination scheme* Φ^L to be such that it satisfies $\Phi_G^L(1), \Phi_G^L(2), \dots, \Phi_G^L(n_1)$ are the $v \in V$ such that $\hat{b}(v) = 1$, listed in increasing order of the geometric mean in Equation (8), and $\Phi_G^L(n_1 + 1), \Phi_G^L(n_1 + 2), \dots, \Phi_G^L(n)$ are the $v' \in V$ such that $\hat{b}(v') \neq 1$, listed in decreasing order of the geometric mean in Equation (9).

4.2 Solving the seeded graph matching problem

In this section we discuss how to compute \hat{b} in the likelihood maximization vertex nomination scheme Φ^L defined in the previous section.

Given any $A, B \in \mathbb{R}^{(m+n) \times (m+n)}$, the *quadratic assignment problem* is to minimize $\|A - PBP^T\|_F^2$ over all permutation matrices $P \in \{0, 1\}^{(m+n) \times (m+n)}$, where $\|\cdot\|_F$ denotes the Frobenius matrix norm. If A and B are respectively adjacency matrices for two graphs, then this is called the *graph matching problem*; it is clearly equivalent to finding a bijection from the vertex set of one graph to the vertex set of the other graph so as to minimize the number of adjacency disagreements induced by the bijection. If P is further constrained so that the upper left corner is the $m \times m$ identity matrix, then the problem is called the *seeded quadratic assignment problem*/ *seeded graph matching problem*; for graphs, this further restriction just means that part of the bijection between the vertex sets is fixed.

Note that the objective function can be simplified (under the restriction that P is a permutation matrix) as $\|A - PBP^T\|_F^2 = \|A\|_F^2 + \|B\|_F^2 - 2\langle A, PBP^T \rangle$, where $\langle \cdot, \cdot \rangle$ is the usual inner product $\langle C, D \rangle := \sum_{i,j} C_{ij}D_{ij}$. Thus the above problems can be phrased as maximize $\langle A, PBP^T \rangle$ over all permutation matrices P .

The optimization problem in Equation (7), for which \hat{b} is the solution, is precisely the seeded quadratic assignment problem above, where $A \in \mathbb{R}^{(m+n) \times (m+n)}$ is the adjacency matrix for the graph G , that is $A_{i,j} := \delta_{i \sim_G j}$ for all $i, j \in W \equiv \{1, 2, \dots, m+n\}$, and $B \in \mathbb{R}^{(m+n) \times (m+n)}$ is the matrix wherein $B_{i,j} := \log \left(\frac{\Lambda_{\mathbf{b}'(i), \mathbf{b}'(j)}}{1 - \Lambda_{\mathbf{b}'(i), \mathbf{b}'(j)}} \right)$ for all $i, j \in W$, where \mathbf{b}' is the member of \mathfrak{B} for which the sequence $\mathbf{b}'(m+1), \mathbf{b}'(m+2), \dots, \mathbf{b}'(m+n)$ are 1's contiguously, then 2's contiguously, \dots , then K 's contiguously. The $\mathbf{b} \in \mathfrak{B}$ —over which the objective function in Equation (7) is maximized—correspond precisely to the permutation matrices P in the seeded quadratic assignment problem, where the upper left corner of P is restricted to be the $m \times m$ identity matrix. We will call this problem a seeded graph matching problem because A is an adjacency matrix. (And we can also choose to think of B as an weighted adjacency matrix for a graph.)

The seeded graph matching problem is computationally hard; indeed, the quadratic assignment problem is NP-hard, and even deciding if two graphs are isomorphic is notoriously of unknown

complexity [12], [23]. However, approximate solutions can be found efficiently with the SGM (Seeded Graph Matching) Algorithm of [16], which is a seeded version of the FAQ algorithm of [25]. (Indeed, SGM is more effective than convex relaxation techniques, as was recently shown in [15].) We employ the SGM algorithm to obtain an approximate solution to \hat{b} for use in the likelihood maximization vertex nomination scheme. It runs in time $O(n^3)$, and can be implemented even when n is approximately 1000.

5 The spectral partitioning vertex nomination scheme

In this section we introduce the *spectral partitioning vertex nomination scheme*. Suppose G is distributed $\text{SB}(K, m, n, b, \Lambda)$. We do not need to assume here that we know n_1, n_2, \dots, n_K , nor the entries of Λ ; we just need to know the value of K and $d := \text{rank of } \Lambda$. (Indeed, by the results in [8], even just knowing an upper bound on d will be sufficient to obtain good performance.)

Say that the adjacency matrix for G is $A \in \{0, 1\}^{(m+n) \times (m+n)}$, that is $A_{i,j} := \delta_{i \sim_G j}$ for all $i, j \in W \equiv \{1, 2, \dots, m+n\}$. Compute d eigenvectors associated, respectively, with the d largest-modulus eigenvalues of A . Scale these eigenvectors so that their respective lengths are the square roots of the absolute values of their corresponding eigenvalues, and define $X \in \mathbb{R}^{(m+n) \times d}$ to have these scaled eigenvectors as its respective columns. The rows of X are low-dimensional embeddings of the corresponding vertices. Now, cluster the rows of X into K clusters; i.e. solve the problem minimize $\|X - C\|_F$ over all matrices $C \in \mathbb{R}^{(m+n) \times d}$ with the property that each row of C is equal to one of just K row vectors, and the values of these K row vectors are also variables to be optimized over.

Say that c is the most frequent value of row vector in the optimal C among the rows corresponding to the vertices $\{u \in U : b(u) = 1\}$. (In other words, c is the centroid associated with the most vertices known to be in the first block.) The spectral partitioning vertex nominating scheme, denote it by Φ^S , associates with G the ordering (of vertices in V) $\Phi_G^S(1), \Phi_G^S(2), \dots, \Phi_G^S(n)$ in increasing order of Euclidean distance between c and their corresponding row in X .

Suppose we consider a sequence of graphs realized from the distributions $\text{SB}(K, m, n, b, \Lambda)$ for, successively, $m+n = 1, 2, 3, \dots$, where K and Λ are fixed, and Λ is positive semi-definite with the property that no two of its rows are equal. Also, assume that $m_1 \geq 1$, and there exists a positive constant γ such that, for all $i = 1, 2, \dots, K$, it holds that $m_i + n_i \geq \gamma(m+n)^{\frac{3}{4}+\gamma}$. It was recently shown in [18] (following the work in [24] and [8]) that almost surely there are no incorrectly clustered vertices in the limit. This implies that the mean average precision of Φ^S converges to 1 as $m+n \rightarrow \infty$.

It will be computationally convenient to approximately (but very quickly) solve the clustering subproblem. This approximate clustering can be done with the (computationally simple) k-means algorithm or with the more general mclust procedure[10, 11]. In both cases, the vertices are nominated based on distance to cluster centroids; in k-means this amounts to the usual Euclidean distance, while for mclust this amounts to nominating based on the Mahalanobis distance. We used mclust for the simulations and real-data experiments in this paper, since empirically it nominated better than k-means.

6 Simulations: Comparing the vertex nomination schemes at three different scales

In this section, we compare and contrast these vertex nomination schemes using three simulation experiments—essentially the same experiment at three different scales, “small scale,” “medium scale,” and “large scale.” For each of the three experiments, we have $K = 3$ blocks in the stochastic block model. The matrix of Bernoulli parameters Λ is

$$\Lambda(\vartheta) := \vartheta \begin{bmatrix} .5 & .3 & .4 \\ .3 & .8 & .6 \\ .4 & .6 & .3 \end{bmatrix} + (1 - \vartheta) \begin{bmatrix} .5 & .5 & .5 \\ .5 & .5 & .5 \\ .5 & .5 & .5 \end{bmatrix}$$

with the value $\vartheta = 1$ for the small scale experiment, $\vartheta = 0.3$ for the medium scale experiment, and $\vartheta = 0.1$ for the large scale experiment, in order to decrease the signal when the number of vertices is larger.

Specifically, the matrix Λ for the small scale experiment, for the medium scale experiment, and for the large scale experiment are, respectively,

$$\Lambda(1) = \begin{bmatrix} .5 & .3 & .4 \\ .3 & .8 & .6 \\ .4 & .6 & .3 \end{bmatrix}, \quad \Lambda(0.3) = \begin{bmatrix} .50 & .44 & .47 \\ .44 & .59 & .53 \\ .47 & .53 & .44 \end{bmatrix}, \quad \Lambda(0.1) = \begin{bmatrix} .50 & .48 & .49 \\ .48 & .53 & .51 \\ .49 & .51 & .48 \end{bmatrix},$$

so that as the number of vertices increases we have that ϑ gets closer to zero, which means that the blocks become less and less stochastically differentiable one from the other. Another notable feature of the Λ here is that the block of interest—the first block—is the intermediate density block; i.e. the Bernoulli adjacency parameter for vertices in the first block is between the Bernoulli adjacency parameter for vertices in the second block and in the third block. This makes it more challenging to identify the vertices of the first block, which is the block of interest.

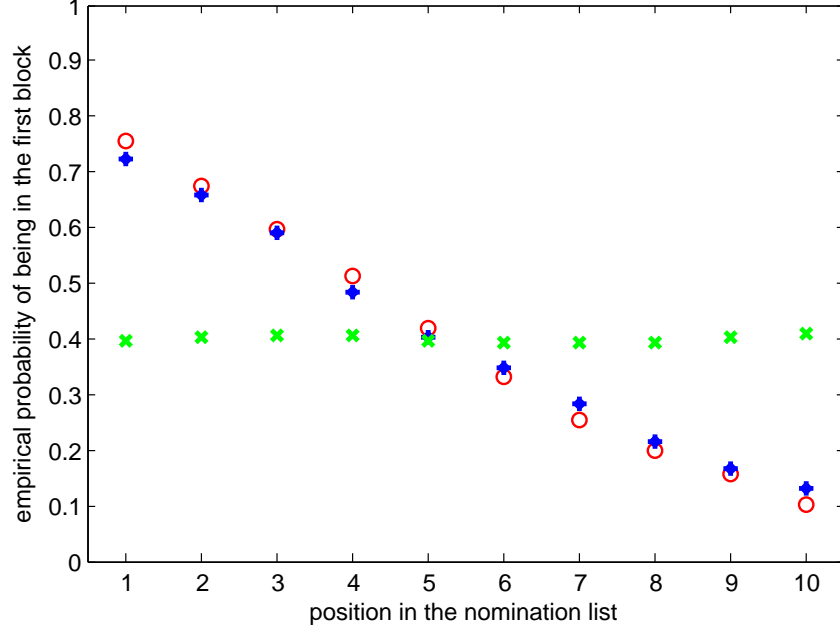
The values of (n_1, n_2, n_3) are taken to be multiples of $(4, 3, 3)$; specifically, in the small scale experiment $(n_1, n_2, n_3) = (4, 3, 3)$, in the medium scale experiment $(n_1, n_2, n_3) = (200, 150, 150)$, and in the large scale experiment $(n_1, n_2, n_3) = (4000, 3000, 3000)$. As for the seeds, the values of (m_1, m_2, m_3) in the respective experiments were taken as $(4, 0, 0)$, $(20, 0, 0)$, and $(40, 0, 0)$.

These three experiments were performed as follows. We independently realized 50000 graphs from the associated distribution of the small scale experiment, 200 graphs in the medium range experiment, and 100 graphs in the large scale experiment. To each observed graph we applied each of: the canonical vertex nomination scheme Φ^C , the likelihood maximization vertex nomination scheme Φ^L , and the spectral partitioning vertex nomination scheme Φ^S . Then, for each vertex nomination scheme, we recorded the fraction of the realizations for which the first nominee of the nomination list was a member of the block of interest, the fraction of the realizations for which the second nominee was a member of the block of interest,..., the fraction of the realizations for which the n th nominee was a member of the block of interest. In Figure 1a, Figure 1b, and Figure 1c these empirical probabilities are plotted against nomination list position, for the three respective experiments and nomination schemes.

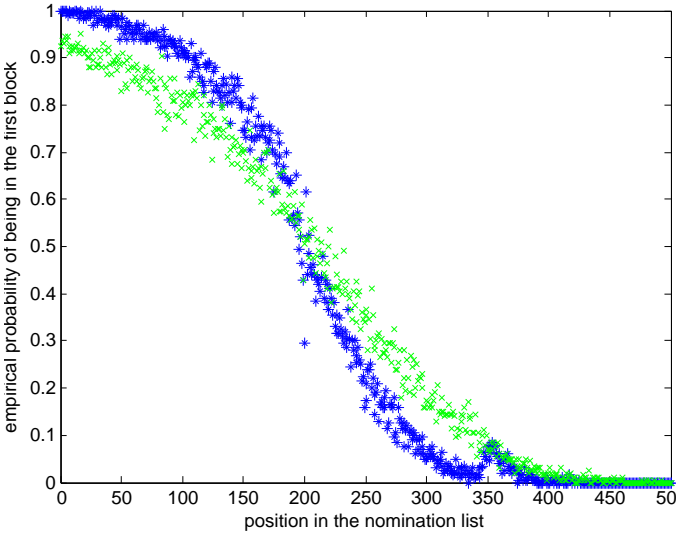
In the small scale experiment, where $n = 10$, the likelihood maximization nomination scheme performed about as well as the (“gold standard”) canonical nomination scheme, and the spectral partitioning nomination scheme performed very poorly—near chance. Then, in the medium scale experiment where $n = 500$, the canonical nomination scheme was no longer practical to compute, and the spectral partitioning nomination scheme performed nearly as well as the likelihood maximization nomination scheme. For a few thousand vertices it was not practical to implement the likelihood maximization nomination scheme, so in the large scale experiment, where $n = 10000$, the only nomination scheme that could be implemented was the spectral partitioning nomination scheme.

The empirical mean average precision for the canonical, likelihood maximization, and spectral partitioning vertex nomination schemes in the three experiments were as follows: (Note that the mean average precision for chance is .4.)

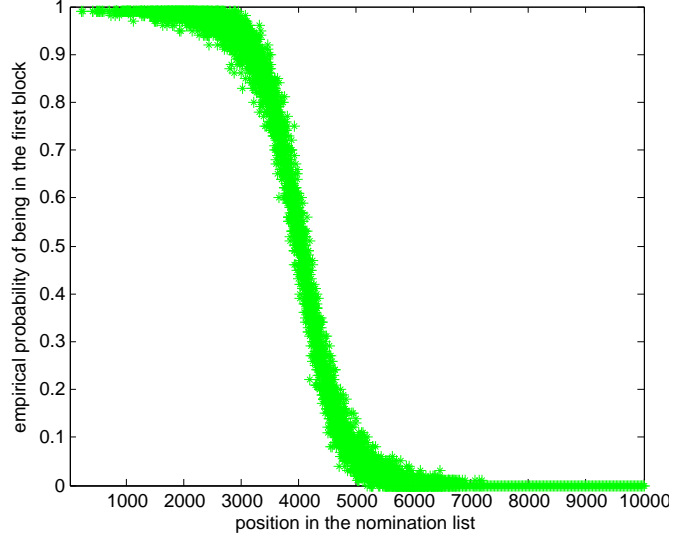
Mean Average Precision	canonical	likelihood-max	spectral part.
small scale exper., $n = 10, \vartheta = 1$.6953	.6716	.3999
medium scale exper., $n = 500, \vartheta = 0.3$	*	.9541	.8683
large scale exper., $n = 10000, \vartheta = 0.1$	*	*	.9901



(a) Small-scale; $n = 10$



(b) Medium-scale; $n = 500$



(c) Large-scale $n = 10,000$

Figure 1: The canonical vertex nomination scheme is in red, the likelihood maximization vertex nomination scheme is in blue, and the spectral partitioning vertex nomination scheme is in green. Note: The canonical vertex nomination scheme is not shown in the medium-scale experiment, nor are the canonical and likelihood maximization vertex nomination schemes shown in the large scale experiment; they are not remotely practical to compute for their respective values of n .

The running times were as follows:

Running Time per simulation	canonical	likelihood-max.	spectral part.
small scale experiment, $n = 10$	≈ 1.4 seconds	$\approx .04$ seconds	$\approx .02$ seconds
medium scale experiment, $n = 500$	*	≈ 286 seconds	$\approx .8$ seconds
large scale experiment, $n = 10000$	*	*	≈ 534 seconds

Indeed, each of the three vertex nomination schemes is superior (in the sense of effectiveness, given practical computability limitations) to the other two at one of the three scales. At a small scale you should use canonical, at a medium scale you should use likelihood maximization, and at a large scale you should use spectral partitioning vertex nomination scheme.

7 Real data examples

While the stochastic block model is often useful for modeling real data, many times real data does not fit the model particularly well. In the following real-data experiments we see that the likelihood maximization vertex nomination scheme is robust to the pathologies inherent to real data.

7.1 Example: The Enron Graph

The Enron Corporation was a highly regarded, large energy company that went spectacularly bankrupt in the early 2000's amid systemic internal fraud. Enron has since become a popular exemplar of corporate fraud and corruption. In the wake of Enron's collapse, the US Energy Regulatory Commission collected a corpus of more than 600,000 emails sent between Enron employees, and this corpus was made public by the US Department of Justice and is available online at a number of websites, including <http://research.cs.queensu.ca/home/skill/siamworkshop.html>.

In [22], the authors restrict their attention to a 189 week period from the year 1998 through the year 2002; they identify 184 distinct email addresses in the Enron email corpus over this time interval, and they identify the pairs of these email addresses that had email communication with each other. Our "Enron Graph" that we use here is based on the graph in [22]; our vertex set W consists of the 128 active email addresses for which the employee's job title in Enron was known. For every pair of such vertices, the pair of vertices are declared adjacent to each other when there was at least one email sent from either of the email addresses to the other. We then divided the

vertices into two blocks: The “upper-echelon” set of vertices $\{w \in W : b(w) = 1\}$ are the vertices whose job titles were designated as CEO, president, vice president, chief manager, company attorney, and chief employee. The “lower-echelon” set of vertices $\{w \in W : b(w) = 2\}$ are the vertices whose job titles were designated as employee, employee administrative, specialist, analyst, trader, director, and manager (besides chief manager, which we designated upper echelon). We chose to group the job titles of manager and director with lower-echelon because a by-eye assessment of the graph indicated that their adjacency affinity was closer to the rest of the lower-echelon vertices. Indeed, this graph is certainly not a realization of an actual two-block stochastic block model, but for the purpose of illustration we will view it as very roughly having some two-block structure. The graph is pictured in Figure 2, as rendered by the standard graph drawing tool in igraph, which can be found at <http://cran.r-project.org/web/packages/igraph/index.html>.

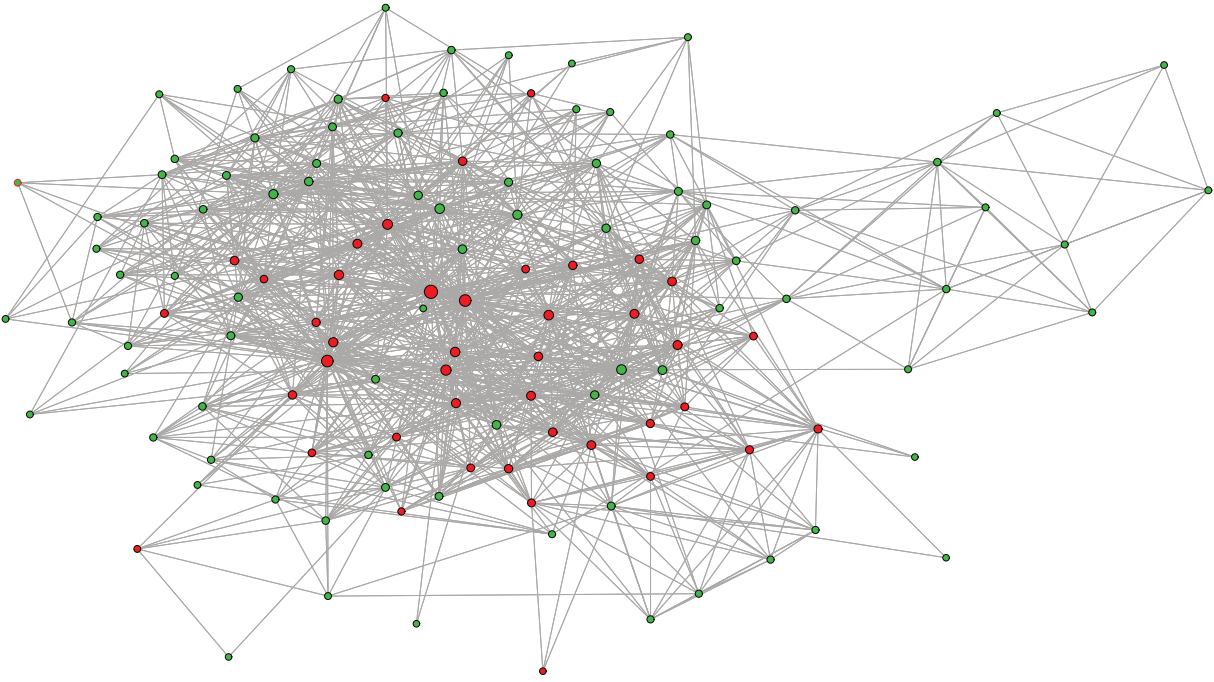


Figure 2: The Enron Graph, as rendered by the standard graph drawing package in igraph. Upper-echelon vertices are red, lower-echelon vertices are green.

We consider the follow experiment. From the 43 upper-echelon vertices $\{w \in W : b(w) = 1\}$, discrete-uniform randomly select $m_1 = 10$ to have their block labels known, and the remaining $n_1 = 33$ have their block labels obscured. From the 85 lower-echelon vertices $\{w \in W : b(w) = 2\}$, independently, discrete-uniform randomly select $m_2 = 20$ to have their block labels known, and the remaining $n_2 = 65$ have their block labels obscured. Then compute $\hat{\Lambda}_{1,1}$, $\hat{\Lambda}_{2,2}$, and $\hat{\Lambda}_{1,2}$ as, respectively, the number of edges in the graph induced by the known upper-echelon vertices, the

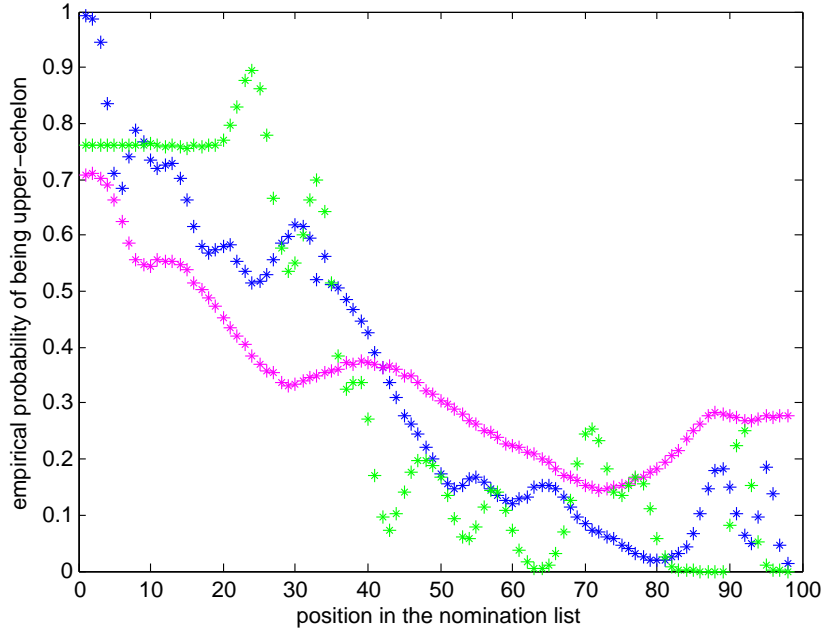


Figure 3: Vertex nomination in the Enron Graph. The likelihood maximization, spectral partitioning, and OTS vertex nomination schemes are, respectively, in blue, green, and purple.

number of edges in the graph induced by the known lower-echelon vertices, and the number of edges in the bipartite graph induced by the known upper-echelon and the known lower-echelon vertices, divided respectively by $\binom{n_1}{2}$, $\binom{n_2}{2}$, and $n_1 n_2$. Then perform likelihood maximization and spectral partitioning vertex nomination on this graph, using $\hat{\Lambda}$ as an substitute for Λ .

We independently repeated this experiment 30000 times; Figure 3 plots the empirical probabilities of vertex membership in the upper echelon for the respective 98 positions in the nomination list, using the likelihood maximization vertex nomination scheme (in blue) and the spectral partitioning vertex nomination schemes (in green). We also included a vertex nomination scheme OTS (in purple), which is described and discussed later in Section 8. These three vertex nomination schemes had empirical mean average precisions .7779 (likelihood maximization), .7619 (spectral partitioning), and .5970 (OTS). For comparison, the mean average precision of chance is .3367.

Note here that the overall classification success of spectral partitioning (i.e. the nominating success averaged over the first 33 positions of the nomination list) is seen in Figure 3 as being comparable to the classification success of likelihood maximization. Also, here the mean average precision of spectral partitioning nomination is comparable to that of likelihood maximization nomination. However, here, very near the top of the nomination list, there is a visible plateau in the spectral partitioning nomination success, whereas maximum-likelihood is nominating very

well; indeed, the first few nominees are almost always from the block of interest.

7.2 Example: The *Caenorhabditis elegans* connectome

The *Caenorhabditis elegans* (C.elegans) is a small roundworm whose connectome (neural-wiring) has been completely mapped out; see <http://www.openconnectomeproject.org/#!/celegans/c5tg>. Our graph here has vertex set W consisting of the 253 non-isolated neurons and, for every pair of vertices, the two vertices are defined to be adjacent to each other if they are adjoined by a chemical synapse. Each neuron (i.e. vertex) is exactly one of the following neuron types: motor neuron, interneuron, or sensory neuron. For each $w \in W$, we define the block membership $b(w)$ to be 1, 2, 3 respectively according as the neuron is a motor neuron (there are 110 of these), interneuron (there are 76 of these), or sensory neuron (there are 67 of these). The graph is pictured in Figure 4 as rendered by the standard graph drawing tool in igraph.

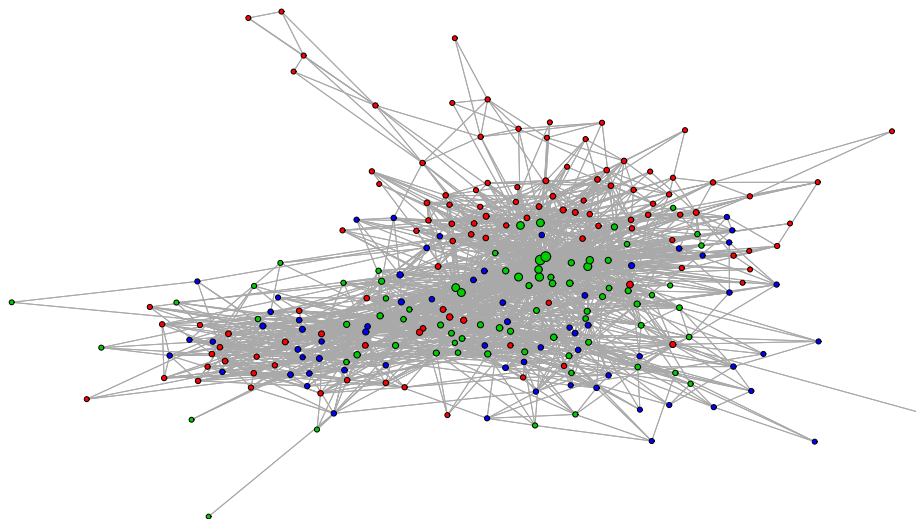


Figure 4: The C. Elegans connectome graph, as drawn by igraph; the motor neurons are colored red, the interneurons are colored green, and the sensory neurons are colored blue.

Consider the following experiment. Block membership is revealed for 30 discrete-uniformly selected motor neurons, 20 discrete-uniformly selected interneurons, and 10 discrete-uniformly selected sensory neurons. We are interested in forming a nomination list out of the remaining 193 ambiguous neurons so that the beginning of the nomination list has an abundance of (the remaining 80) ambiguous motor neurons.

Perhaps the story behind your desire for this nomination list might be that you wish to study motor neurons, but have limited resources to biochemically test neuron type for the ambiguous neu-

rons. The nomination list would be used to order the ambiguous neurons for the testing, to identify as many motor neurons as possible from the ambiguous neurons before your resources are depleted.

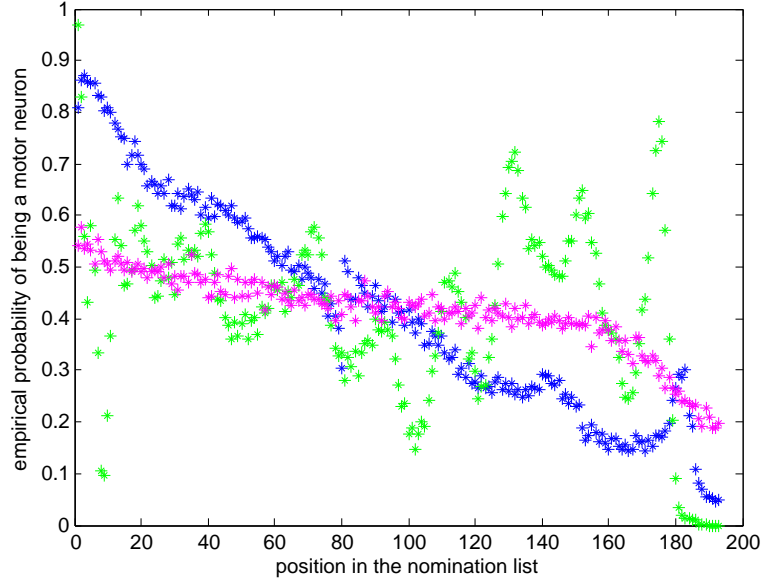


Figure 5: Vertex nomination for motor neurons in *C. Elegans*: Likelihood maximization is colored blue, spectral partitioning is colored green, and OTS is colored purple.

We repeated this experiment 1000 times, each time nominating for motor neurons using the likelihood maximization and the spectral partitioning vertex nomination schemes. We also used the OTS vertex nomination scheme described later in Section 8. In each repetition, we estimated Λ with $\hat{\Lambda}$, whose entries reflect the edge densities in the subgraphs induced by the various blocks intersecting the seeds. The empirical mean average precision for the likelihood maximization, spectral partitioning, and OTS vertex nomination schemes were respectively 0.7272, 0.5096, and 0.5041; the mean average precision of chance is 0.4145. Figure 5 shows that empirical probability of being a motor neuron at every position in the vertex nomination list, for the likelihood maximization (blue), spectral partitioning (green) and OTS (purple) vertex nomination schemes. (The OTS vertex nomination scheme is defined and described later in Section 8.)

Note that here spectral partitioning performed very erratically and (overall) poorly. This might be attributed to a lack of our idealized three-block-structure here; that is to say, this graph doesn't appear to be an instantiation of monolithic stochastic behavior for vertices within the respective three blocks. In this case here, likelihood maximization is seen to be more robust to the lack of idealized block model setting, and still maintained a steady and very pronounced slope in Figure 5.

7.3 Example: A political blog network

The political blogosphere data in our next example was collected in [1] around the time of the US presidential election in 2004. This data set consists of 1224 weblogs (“blogs”), each of which web-links to—or is web-linked from—at least one other of these blogs. These blogs form the vertex set W of our graph. Each of the blogs was classified by [1] as being either liberal or conservative; for each $w \in W$ we define $b(w)$ to be 1 or 2, according as w was classified liberal or conservative. There are 588 liberal blogs and 636 conservative blogs here. For each pair of vertices/ blogs, the pair are adjacent if at least one of the blogs links to the other. In Figure 6 this graph is drawn using igraph.

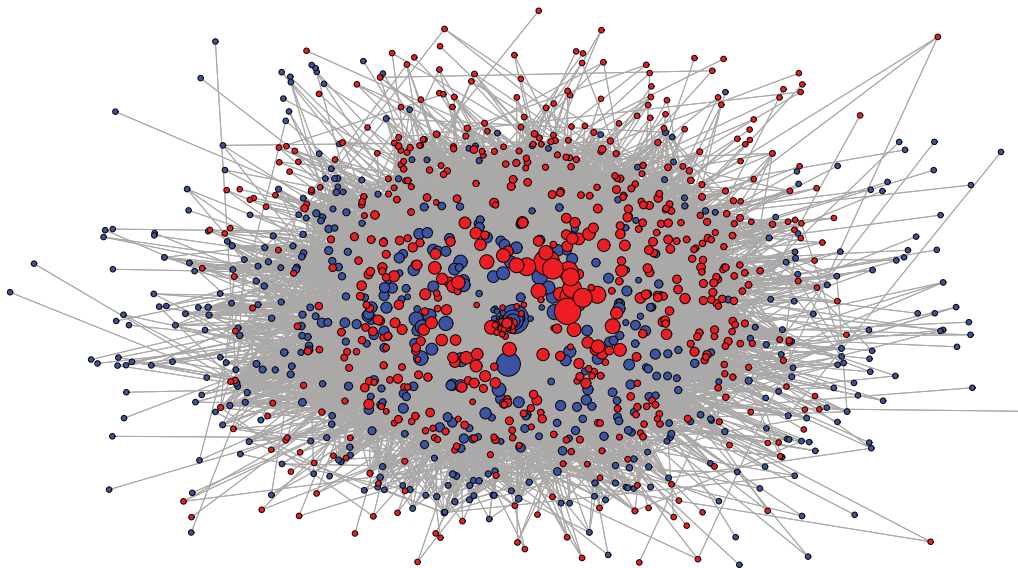


Figure 6: The political blogosphere graph, as drawn by igraph; liberal blogs/vertices are colored blue and conservative blogs/vertices are colored red.

Consider the following experiment. Discrete-uniform-randomly select 80 liberal and 80 conservative blogs to have their political orientation revealed, and create a nomination list for the remaining 1064 ambiguous blogs. The story could be that you work for a political action committee, and want to make a report summarizing liberal blog views on some current event. You have a limited amount of blog-reading time, and only know the content and political affiliations of a few of the blogs. Thus, you want to create a nomination list which will provide the order for your reading the ambiguous blogs, so that you read many liberal blogs in your limited time.

We repeated this experiment 1000 times, and calculated the likelihood maximization, spectral partitioning, and OTS vertex nomination schemes for each repetition. See the results in Figure 7.

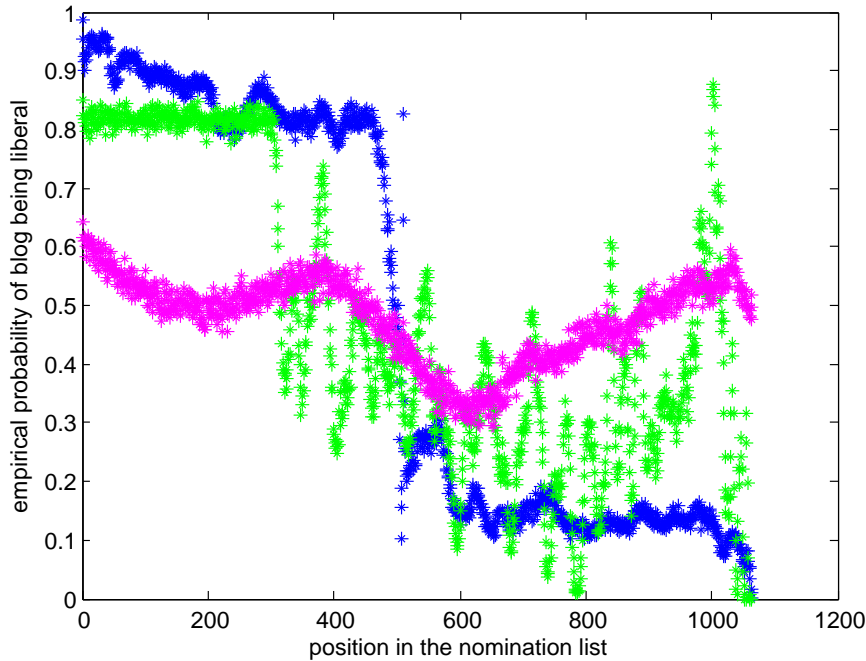


Figure 7: Nominating blogs; likelihood maximization vertex nomination scheme is colored blue, spectral partitioning vertex nomination scheme is colored green, OTS is colored purple.

The mean average precision for the likelihood maximization, spectral partitioning, and OTS vertex nomination schemes were, respectively, .8922, .7856, and .5429; the mean average precision for chance nomination is .4774.

7.4 Example: A movie network

The movie data set for this section was created by scraping movie infoboxes from Wikipedia. We examined all movies released between the year 2000 and the year 2010 from any of the five movie studios 20th Century Fox, Columbia Pictures, Paramount, Universal, and Warner Brothers. For each of these movies, we recorded the directors, producers, and actors from the Wikipedia infobox, along with the movie genre.

The vertex set W of our Movie Graph is taken to be the set of above-mentioned movies which belong to exactly one genre out of comedy, action thriller, and drama. For each $w \in W$, $b(w)$ is defined to be 1, 2, 3 according as the movie was classified comedy, action thriller, or drama. There were 227 comedies, 157 action thrillers, and 235 drama movies, so the number of vertices in the graph is 619. For any pair of vertices in W , we declare them to be adjacent if they have a director, producer, or actor in common. Figure 8 illustrates the graph, as drawn by igraph.

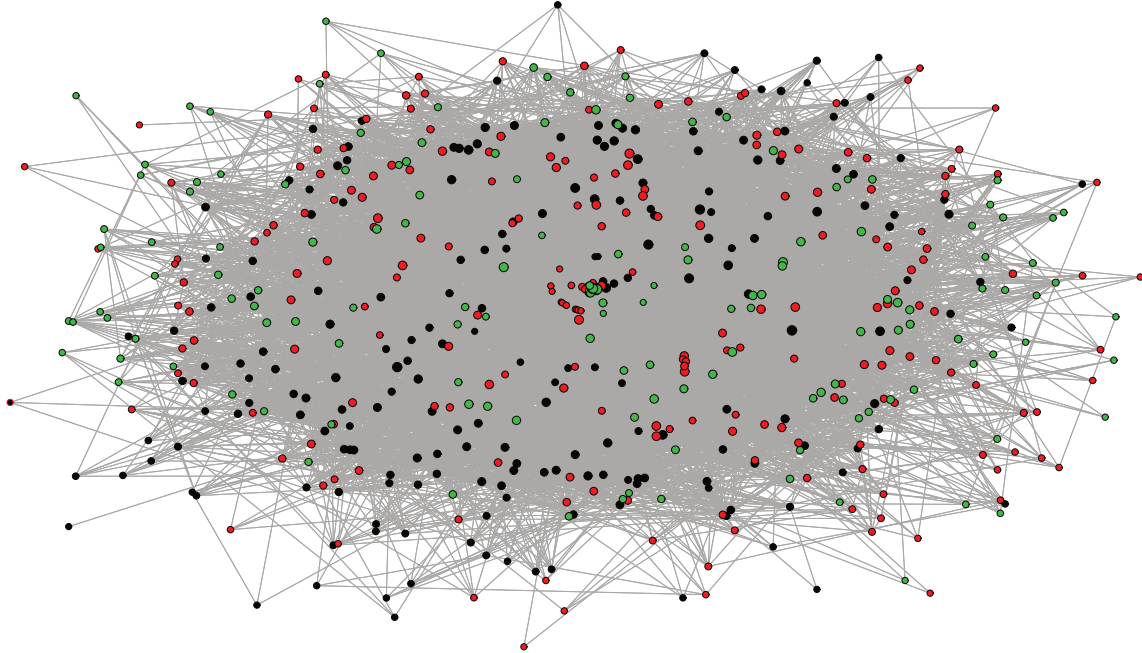


Figure 8: The movie graph, as drawn by igraph; comedies are colored black, action thrillers are colored green, and dramas are colored red.

Consider the experiment where we discrete-uniform randomly select 30 vertices from each block to serve as seeds, which leaves 529 ambiguous vertices. We then seek a nomination list which nominates the movies to be comedies. Indeed, the story might be that you have a limited amount of time to watch movies, and want to get some laughs. Such a nomination list would tell you the order to watch the movies in, until you run out of movie-watching time.

We performed 1000 repetitions of this experiment. The mean average precision for the likelihood maximization, spectral partitioning, and OTS vertex nomination schemes were, respectively, .5814, .3764, and .3766. The latter two performed approximately the same as chance, which has a mean average precision of .3724. Figure 9 shows a plot for the empirical probability of being a comedy for the different positions in the nomination list under different vertex nomination schemes.

Note that here the erratic behavior of spectral partitioning nomination rendered it effectively useless. There seems to be very significant deviations from the idealized stochastic three-block model, and there seems to be less-than-subtle sub-block structure that threw off the spectral machinery from separating out the gross three-block structure we hypothesized. In addition, the movie genre field in the Wikipedia infobox does not seem much more than a vague characterization, since the dividing lines between genre are blurred and somewhat arbitrary. Nonetheless,

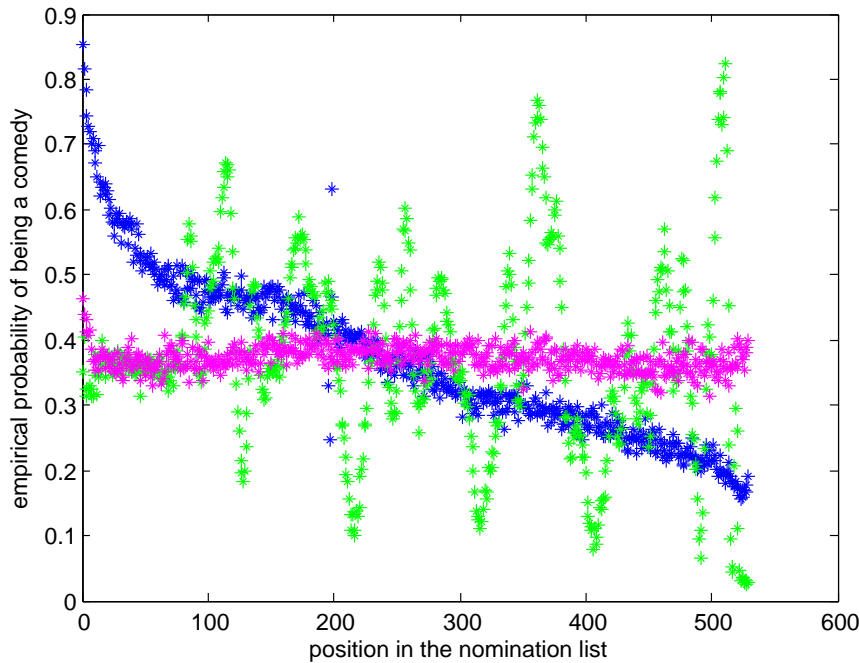


Figure 9: The movie graph; likelihood maximization vertex nomination scheme is colored blue, spectral partitioning vertex nomination scheme is colored green, OTS is colored purple.

here likelihood maximization is showing much robustness to the lack of idealized conditions hypothesized, and the top of its nominating list is showing a pronounced early slope. In all of these real-data experiments of Sections 7.1, 7.2, 7.3, and 7.4, where the number of ambiguous vertices ranged from around 100 to around 1000, likelihood maximization was the best vertex nomination scheme among those we tried.

8 Discussion

In this paper, the currently-popular stochastic block model setting enables the principled development of vertex nomination schemes. We introduced three vertex nomination schemes; the canonical, likelihood maximization, and spectral partitioning vertex nomination schemes. In Section 6 we compared and contrasted the effectiveness and runtime of these three vertex nomination schemes at small, medium, and large scales. In Proposition 1 we proved that the canonical vertex nomination scheme has maximum possible mean average precision among all vertex nomination schemes, and thus it should be used as long as it is computationally feasible, which is up to a few tens of vertices. (The runtime visibly grows exponentially in the number of vertices.) The

likelihood maximization vertex nomination scheme, which utilizes state of the art graph matching machinery, should be used next (i.e. when the canonical vertex nomination scheme can't be used), as long as it is computationally feasible, which is up to around 1000 or 1500 vertices. Sections 7.1, 7.2, 7.3, and 7.4 then feature illustrations with real data, and illustrate robustness of maximum-likelihood nomination to model pathology inherent in real data.

These vertex nomination schemes are simple, yet effective. The likelihood maximization and spectral partitioning vertex nomination schemes are grown from basic block estimation strategies. Going forward, we expect to see the next generation of vertex nomination schemes build on similar such adaptations of block estimation strategies. For an excellent survey of the literature on community detection in networks—including the setting of stochastic block models—and available algorithms, see the very comprehensive survey article of Fortunato [9] and papers cited therein, such as Newman and Girvan [19] and the classic article of Nowicki and Snijders [20]. Also see Latent Dirichlet Allocation (LDA) [3] of Blei, Ng, and Jordan. Under-the-hood modifications of existing community detection algorithms, including LDA and LDA-based methodologies, will yield new vertex nomination schemes that will be increasingly effective and fast. We also expect even more effective vertex nomination schemes to come from merging vertex nomination techniques, perhaps similar in spirit to the work in [17], where graph matching and spectral partitioning are merged into a more effective avenue of graph matching for large graphs.

Lastly, it is worth mentioning that we also took state of the art, off-the-shelf block estimation R code from <http://cran.r-project.org/web/packages/lda/lda.pdf> based on the article [2], and used it on our real data from this paper. We produced a nomination order based on that algorithm's outputted estimated parameter of the Dirichlet priors per-vertex block distribution (and the vertices with known block labels served the purpose of identifying which estimated block was the block of interest). We called this vertex nomination scheme "OTS" ("Off the Shelf"). Indeed, the block estimation algorithm was not specifically designed for vertex nomination; we used it off the shelf, and some under the hood modification are expected to yield substantial improvement.

Acknowledgements: The authors thank Daniel Sussman and Stephen Chestnut for valuable discussions, and an anonymous referee for very thoughtful suggestions that greatly improved the paper. This work is partially supported by Johns Hopkins University Human Language Technology Center of Excellence (JHU HLT COE) and the XDATA program of the Defense Advanced Research Projects Agency (DARPA) administered through Air Force Research Laboratory contract FA8750-12-2-0303.

References

- [1] L.A. Adamic and N. Glance, The political blogosphere and the 2004 US election: divided they blog. *ACM: Proceedings of the 3rd International Workshop on Link Discovery* (2005), pp 36–43.
- [2] E.M. Airoldi, D.M. Blei, S.E. Fienberg, E. Xing, Mixed membership stochastic blockmodels, *Advances in Neural Information Processing Systems* (2009), pp 33–40.
- [3] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, *Journal of Machine Learning Research* **3** (2003), pp 993–1022.
- [4] P.J. Bickel and A. Chen, A nonparametric view of network models and Newman-Girvan and other modularities, *Proceedings of the National Academy of Sciences of the United States* **106:50**, (2009), pp 21068–21073.
- [5] D. Conte, P. Foggia, C. Sansone, and M. Vento, Thirty years of graph matching in pattern recognition, *International Journal of Pattern Recognition and Artificial Intelligence*, **18** (2004), pp 265–298.
- [6] G.A. Coppersmith and C.E. Priebe, Vertex nomination via content and context, submitted. Copy available at <http://arxiv.org/pdf/1201.4118> .
- [7] P. Erdős and A.R. Rényi, Asymmetric graphs, *Acta Math. Acad. Sci. Hungar.*, **14** (1963), pp 295–315.
- [8] D.E. Fishkind, D.L. Sussman, M. Tang, J.T. Vogelstein, and C.E. Priebe, Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown, *SIAM J. Matrix Anal. and Appl.* **34** (2013), pp 23–39.
- [9] S. Fortunato, Community detection in graphs, *Physics Reports*, **486:3** (2010), pp 75–174.
- [10] C. Fraley and A.E. Raftery, MCLUST: Software for model-based cluster analysis, *Journal of Classification*, **16** (1999) 297–306.
- [11] C. Fraley and A.E. Raftery, Enhanced software for model-based clustering, density estimation, and discriminant analysis: MCLUST, *Journal of Classification*, **20** (2003) 263–286.
- [12] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman (1979).

- [13] D.S. Lee and C.E. Priebe, Bayesian vertex nomination, submitted. Copy available at <http://arxiv.org/pdf/1205.5082>.
- [14] A. Lubiw, Some NP-complete problems similar to graph isomorphism, *SIAM Journal on Computing* **10** (1981), pp 11-21.
- [15] V. Lyzinski, D.E. Fishkind, M. Fiori, J.T. Vogelstein, C.E. Priebe, G. Sapiro, Graph Matching: Relax at your own risk, submitted. Copy available at <http://arxiv.org/abs/1405.3133>.
- [16] V. Lyzinski, D.E. Fishkind, and C.E. Priebe, Seeded graph matching for correlated Erdős-Rényi graphs, accepted for publication in *Journal of Machine Learning Research*. Copy available at <http://arxiv.org/abs/1304.7844>.
- [17] V. Lyzinski, D.L. Sussman, D.E. Fishkind, H. Pao, C.E. Priebe, Spectral clustering for divide-and-conquer graph matching, submitted. Copy available at <http://arxiv.org/abs/1310.1297v3>.
- [18] V. Lyzinski, D.L. Sussman, M. Tang, A. Athreya, C.E. Priebe, Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding, submitted. Copy available at <http://arxiv.org/abs/1310.0532>.
- [19] M.E.J. Newman and M. Girvan, Finding and evaluating community structure in networks, *Physical Review E*, **69:2** (2004) 026113.
- [20] K. Nowicki and T.A.B. Snijders, Estimation and prediction for stochastic blockstructures, *Journal of the American Statistical Association*, **96:455** (2001), 1077–1087.
- [21] G. Polya, Kombinatorische anzahlbestimmungen fur gruppen, graphen und chemische verbindungen, *Acta. Math*, **68** (1937), pp 145–254.
- [22] C.E. Priebe, J.M. Conroy, D.J. Marchette, and Y. Park, Scan statistics on Enron graphs, *Computational and Mathematical Organization Theory* **11** (2005), pp 229–247.
- [23] R.C. Read and D.G. Corneil, The graph isomorphism disease, *Journal of Graph Theory* **1** (1977), pp 339–363.
- [24] D.L. Sussman, M. Tang, D.E. Fishkind, and C.E. Priebe, A consistent adjacency spectral embedding for stochastic blockmodel graphs, *Journal of the American Statistical Association* **107** (2012), pp 1119–1128.

- [25] J.T. Vogelstein, J.M. Conroy, L.J. Podrazik, S.G. Kratzer, E.T. Harley, D.E. Fishkind, R.J. Vogelstein, and C.E. Priebe, Brain graph matching via fast approximate quadratic programming, submitted. Copy available at arxiv.org/pdf/1112.5507 .
- [26] M. Zaslavskiy, F. Bach, and J.P. Vert, A path following algorithm for the graph matching problem, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31:12** (2009), pp 2227–2242.

Appendix A A limit result related to likelihood maximization vertex nomination scheme

The purpose of this section/appendix is to state and prove Theorem 2. Consider here a sequence of random graphs realized from $\text{SB}(K, m, n, b, \Lambda)$ for, successively, $n = 1, 2, 3, \dots$, where K and Λ are fixed. The values of n_1, n_2, \dots, n_K are each functions of n , and we assume there exists a real number $\gamma > 0$ such that, for all $i = 1, 2, \dots, K$, it holds that $n_i \geq \gamma \cdot n$ for all but a finite number of values of n . The values of m_1, m_2, \dots, m_K are each taken to be 0; i.e. seeds are not required in the context of Theorem 2, thus here we will have $W = V$ and $U = \emptyset$. Now, \mathfrak{B} denotes the set of functions $\mathfrak{b} : W \rightarrow \{1, 2, \dots, K\}$ such that for all $i = 1, 2, \dots, K$, $|\{w \in W : \mathfrak{b}(w) = i\}| = n_i$.

For each $\mathfrak{b} \in \mathfrak{B}$ define

$$f(\mathfrak{b}) := \sum_{\{w, w'\} \in \binom{W}{2}} \delta_{w \sim_G w'} \log \left(\frac{\Lambda_{\mathfrak{b}(w), \mathfrak{b}(w')}}{1 - \Lambda_{\mathfrak{b}(w), \mathfrak{b}(w')}} \right) \quad \text{and} \quad g(\mathfrak{b}) := \sum_{\{w, w'\} \in \binom{W}{2}} \delta_{w \sim_G w'} \Lambda_{\mathfrak{b}(w), \mathfrak{b}(w')}.$$

In the description of the likelihood maximization vertex nomination scheme in Section 4.1, recall that the first step consisted of computing the classifier $\hat{b} = \arg \max_{\mathfrak{b} \in \mathfrak{B}} f(\mathfrak{b})$, which was an estimate of the block membership function b ; in this section/appendix we shall consider the related classifier $\tilde{b} = \arg \max_{\mathfrak{b} \in \mathfrak{B}} g(\mathfrak{b})$. Whereas we did not formally prove the consistency of \hat{b} , we will now prove in Theorem 2 the consistency of \tilde{b} , under mild conditions. Define the *number of first-block misclassifications* $\epsilon := |\{w \in W : b(w) = 1 \text{ and } \tilde{b}(w) \neq 1\}|$. Of course, ϵ is a function of n .

Theorem 2. *With the above assumptions and notation,*

- a) *If $\Lambda_{1,1} \neq \Lambda_{i,j}$ for all $\{i, j\} \neq \{1, 1\}$ then there exists a real number $c > 0$ such that almost surely $\epsilon \leq c \log n$ for all but a finite number of values of n , and*
- b) *If $\Lambda_{i',i'} \neq \Lambda_{i,j}$ for all $\{i, j\} \neq \{i', i'\}$ then almost surely $\epsilon = 0$ for all but a finite number of values of n .*

Proof of Theorem 2, part a: Let us denote $\zeta := \min |\Lambda_{1,1} - \Lambda_{i,j}|$ over all $\{i, j\} \neq \{1, 1\}$.

Let $\mathfrak{b} \in \mathfrak{B}$ be arbitrarily selected. We have

$$g(b) - g(\mathfrak{b}) = \sum_{\{w, w'\} \in \binom{W}{2}} \left(\Lambda_{b(w), b(w')} - \Lambda_{\mathfrak{b}(w), \mathfrak{b}(w')} \right) \delta_{w \sim_G w'}. \quad (10)$$

Next note that, by the definition of \mathfrak{B} ,

$$\sum_{\{w, w'\} \in \binom{W}{2}} \left(\Lambda_{b(w), b(w')} \right)^2 = \sum_{\{w, w'\} \in \binom{W}{2}} \left(\Lambda_{\mathfrak{b}(w), \mathfrak{b}(w')} \right)^2. \quad (11)$$

Taking the expectation in Equation (10), then subtracting and adding half of Equation (11),

$$\begin{aligned}
\mathbb{E}[g(b) - g(\mathbf{b})] &= \sum_{\{w, w'\} \in \binom{W}{2}} \left(\Lambda_{b(w), b(w')} - \Lambda_{\mathbf{b}(w), \mathbf{b}(w')} \right) \Lambda_{b(w), b(w')} \\
&= \sum_{\{w, w'\} \in \binom{W}{2}} \frac{1}{2} \left(\left(\Lambda_{b(w), b(w')} \right)^2 - 2 \cdot \Lambda_{b(w), b(w')} \Lambda_{\mathbf{b}(w), \mathbf{b}(w')} + \left(\Lambda_{\mathbf{b}(w), \mathbf{b}(w')} \right)^2 \right) \\
&= \sum_{\{w, w'\} \in \binom{W}{2}} \frac{1}{2} \left(\Lambda_{b(w), b(w')} - \Lambda_{\mathbf{b}(w), \mathbf{b}(w')} \right)^2.
\end{aligned} \tag{12}$$

Now, for any positive integer s , any independent random variables Z_1, Z_2, \dots, Z_s , and any real numbers $a_1, a_2, \dots, a_s, a'_1, a'_2, \dots, a'_s$ such that, for all $i = 1, 2, \dots, s$, Z_i is interval- $[a_i, a'_i]$ -valued, recall that Hoeffding's Inequality asserts that, for any positive $t \in \mathbb{R}$,

$$\mathbb{P} \left[\left| \sum_{i=1}^s Z_i - \mathbb{E} \sum_{i=1}^s Z_i \right| \geq t \right] \leq 2 \cdot e^{\left(\frac{-2 \cdot t^2}{\sum_{i=1}^s (a'_i - a_i)^2} \right)}. \tag{13}$$

When the Z_i are taken to be the independent random variables summed in Equation (10), note by Equation (12) that $\sum_{i=1}^s (a'_i - a_i)^2 = 2 \cdot \mathbb{E}[g(b) - g(\mathbf{b})]$ here. Thus Hoeffding's inequality here, taking $t := \mathbb{E}[g(b) - g(\mathbf{b})]$, states that

$$\begin{aligned}
\mathbb{P}[g(b) - g(\mathbf{b}) \leq 0] &\leq \mathbb{P} \left[\left| g(b) - g(\mathbf{b}) - \mathbb{E}[g(b) - g(\mathbf{b})] \right| \geq \mathbb{E}[g(b) - g(\mathbf{b})] \right] \\
&\leq 2 \cdot e^{\left(\frac{-2 \mathbb{E}^2[g(b) - g(\mathbf{b})]}{2 \mathbb{E}[g(b) - g(\mathbf{b})]} \right)} = 2 \cdot e^{-\mathbb{E}[g(b) - g(\mathbf{b})]}.
\end{aligned} \tag{14}$$

Define the set $\Omega := \left\{ \{w, w'\} \in \binom{W}{2} : b(w) = b(w') = 1 \text{ and } \mathbf{b}(w) \neq 1 \text{ and } \mathbf{b}(w') \neq 1 \right\}$. Define the set $\Upsilon := \left\{ \{w, w'\} \in \binom{W}{2} : b(w) = b(w') = 1 \text{ and } [\mathbf{b}(w) \neq 1 \text{ exclusive-or } \mathbf{b}(w') \neq 1] \right\}$. Also, define $\epsilon_b := |\{w \in W : b(w) = 1 \text{ and } \mathbf{b}(w) \neq 1\}|$. By Equation (12), we then have that

$$\begin{aligned}
\mathbb{E}[g(b) - g(\mathbf{b})] &\geq \sum_{\{w, w'\} \in \Omega} \frac{1}{2} \left(\Lambda_{b(w), b(w')} - \Lambda_{\mathbf{b}(w), \mathbf{b}(w')} \right)^2 + \sum_{\{w, w'\} \in \Upsilon} \frac{1}{2} \left(\Lambda_{b(w), b(w')} - \Lambda_{\mathbf{b}(w), \mathbf{b}(w')} \right)^2 \\
&\geq \frac{1}{2} \zeta^2 \frac{\epsilon_b(\epsilon_b - 1)}{2} + \frac{1}{2} \zeta^2 (n_1 - \epsilon_b) \epsilon_b \geq \frac{\zeta^2}{5} n_1 \cdot \epsilon_b \geq \frac{\zeta^2 \gamma}{5} n \cdot \epsilon_b
\end{aligned} \tag{15}$$

when n is large enough.

Thus, if we define \mathfrak{B}' to be the set of $\mathbf{b} \in \mathfrak{B}$ such that $\epsilon_b \geq \frac{10}{\zeta^2 \gamma} \log n$, then we have by Equation (14), the fact that $|\mathfrak{B}'| \leq |\mathfrak{B}| \leq n^n$, and subadditivity of the probability measure, that

$$\mathbb{P}[\exists \mathbf{b} \in \mathfrak{B}' : g(b) \leq g(\mathbf{b})] \leq \sum_{\mathbf{b} \in \mathfrak{B}'} 2 \cdot e^{-2n \log n} \leq 2 \cdot e^{n \log n - 2n \log n} = \frac{2}{n^n}$$

which is finitely summable over all n , hence by the Borel-Cantelli Lemma we have almost surely that there are at most a finite number of values of n for which any member of $\arg \max_{\mathbf{b} \in \mathfrak{B}} g(\mathbf{b})$ has at least $\frac{10}{\zeta^{2\gamma}} \log n$ misclassifications, and Theorem 2, part **a**) is shown. \square

Proof of Theorem 2, part b: Let us denote $\tilde{\zeta} := \min |\Lambda_{i',i'} - \Lambda_{i,j}|$ over all $\{i, j\} \neq \{i', i'\}$.

Let $\mathbf{b} \in \mathfrak{B}$ be arbitrarily selected. For each $i = 1, 2, \dots, K$,

define $\Omega_i := \left\{ \{w, w'\} \in \binom{W}{2} : b(w) = b(w') = i \text{ and } \mathbf{b}(w) \neq i \text{ and } \mathbf{b}(w') \neq i \right\}$, and

define $\Upsilon_i := \left\{ \{w, w'\} \in \binom{W}{2} : b(w) = b(w') = i \text{ and } [\mathbf{b}(w) \neq i \text{ exclusive-or } \mathbf{b}(w') \neq i] \right\}$, and

define $\epsilon_{\mathbf{b},i} := |\{w \in W : b(w) = i \text{ and } \mathbf{b}(w) \neq i\}|$. Now, by Equation (12), we have that

$$\begin{aligned} \mathbb{E}[g(b) - g(\mathbf{b})] &\geq \sum_{i=1}^K \left(\sum_{\{w,w'\} \in \Omega_i} \frac{1}{2} \left(\Lambda_{b(w),b(w')} - \Lambda_{\mathbf{b}(w),\mathbf{b}(w')} \right)^2 + \sum_{\{w,w'\} \in \Upsilon_i} \frac{1}{2} \left(\Lambda_{b(w),b(w')} - \Lambda_{\mathbf{b}(w),\mathbf{b}(w')} \right)^2 \right) \\ &\geq \sum_{i=1}^K \left(\frac{1}{2} \tilde{\zeta}^2 \frac{\epsilon_{\mathbf{b},i}(\epsilon_{\mathbf{b},i} - 1)}{2} + \frac{1}{2} \tilde{\zeta}^2 (n_i - \epsilon_{\mathbf{b},i}) \epsilon_{\mathbf{b},i} \right) \geq \sum_{i=1}^K \left(\frac{\tilde{\zeta}^2}{5} n_i \epsilon_{\mathbf{b},i} \right) \geq \frac{\tilde{\zeta}^2 \gamma}{5} n \sum_{i=1}^K \epsilon_{\mathbf{b},i} \end{aligned} \quad (16)$$

when n is large enough.

Now, define \mathfrak{B}'' to be the set of $\mathbf{b} \in \mathfrak{B}$ such that $\sum_{i=1}^K \epsilon_{\mathbf{b},i} \geq \frac{10}{\zeta^{2\gamma}} \log n$. Then, we have by Equation (14), the fact that $|\mathfrak{B}''| \leq |\mathfrak{B}| \leq n^n$, and subadditivity of the probability measure, that

$$\mathbb{P}[\exists \mathbf{b} \in \mathfrak{B}'' : g(b) \leq g(\mathbf{b})] \leq \sum_{\mathbf{b} \in \mathfrak{B}''} 2 \cdot e^{-2n \log n} \leq 2 \cdot e^{n \log n - 2n \log n} = \frac{2}{n^n} \quad (17)$$

Finally, define \mathfrak{B}''' to be the set of $\mathbf{b} \in \mathfrak{B}$ such that $0 < \sum_{i=1}^K \epsilon_{\mathbf{b},i} < \frac{10}{\zeta^{2\gamma}} \log n$. Note that $|\mathfrak{B}'''| \leq \left(\lceil \frac{10}{\zeta^{2\gamma}} \log n \rceil \right) \left(\frac{10}{\zeta^{2\gamma}} \log n \right)! \leq n^{\frac{11}{\zeta^{2\gamma}} \log n} = e^{\frac{11}{\zeta^{2\gamma}} \log^2 n}$, when n is large enough. Thus, by Equation (14) and Equation (16) we have

$$\begin{aligned} \mathbb{P}[\exists \mathbf{b} \in \mathfrak{B}''' : g(b) \leq g(\mathbf{b})] &\leq \sum_{\mathbf{b} \in \mathfrak{B}'''} 2 \cdot e^{-\frac{\tilde{\zeta}^2 \gamma}{5} n} \\ &\leq 2e^{-\frac{\tilde{\zeta}^2 \gamma}{5} n + \frac{11}{\zeta^{2\gamma}} \log^2 n} \leq 2e^{-\frac{\tilde{\zeta}^2 \gamma}{6} n}. \end{aligned} \quad (18)$$

Since the bounds in Equations (17) and (18) are finitely summable over all n , we have by the Borel-Cantelli Lemma that, almost surely, it happens for all but a finite number of values of n that there is no $\mathbf{b} \in \mathfrak{B}$ such that $\sum_{i=1}^K \epsilon_{\mathbf{b},i} > 0$ and $g(b) \leq g(\mathbf{b})$, so Theorem 2, part **b**) is proved. \square