Computational Statistics and Data Analysis 145 (2020) 106916

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Vertex nomination: The canonical sampling and the extended spectral nomination schemes

Jordan Yoder [a], Li Chen [b], Henry Pao [c], Eric Bridgeford [d], Keith Levin [e], Donniell E. Fishkind [f,*], Carey Priebe [f], Vince Lyzinski [g]

[a] Jordan & Yoder, LLC, United States of America
[b] Intel Labs, United States of America
[c] Amazon.com, United States of America
[d] Department of Biostatistics, Johns Hopkins University, United States of America
[e] Department of Statistics, University of Michigan, United States of America
[f] Department of Applied Mathematics and Statistics, Johns Hopkins University, United States of America
[g] Department of Mathematics and Statistics, University of Massachusetts Amherst, United States of America

ARTICLE INFO

Article history:
Received 4 March 2019
Received in revised form 23 December 2019
Accepted 25 December 2019
Available online 20 January 2020

Keywords:
Vertex nomination
Markov chain Monte Carlo
Spectral partitioning
Mclust

ABSTRACT

Suppose that one particular block in a stochastic block model is of interest, but block labels are only observed for a few of the vertices in the network. Utilizing a graph realized from the model and the observed block labels, the vertex nomination task is to order the vertices with unobserved block labels into a ranked nomination list with the goal of having an abundance of interesting vertices near the top of the list. There are vertex nomination schemes in the literature, including the optimally precise canonical nomination scheme $\mathcal{L}^C$ and the consistent spectral partitioning nomination scheme $\mathcal{L}^P$. While the canonical nomination scheme $\mathcal{L}^C$ is provably optimally precise, it is computationally intractable, being impractical to implement even on modestly sized graphs.

With this in mind, an approximation of the canonical scheme – denoted the *canonical sampling nomination scheme* $\mathcal{L}^{CS}$ – is introduced; $\mathcal{L}^{CS}$ relies on a scalable, Markov chain Monte Carlo-based approximation of $\mathcal{L}^C$, and converges to $\mathcal{L}^C$ as the amount of sampling goes to infinity. The spectral partitioning nomination scheme is also extended to the *extended spectral partitioning nomination scheme*, $\mathcal{L}^{EP}$, which introduces a novel semisupervised clustering framework to improve upon the precision of $\mathcal{L}^P$. Real-data and simulation experiments are employed to illustrate the precision of these vertex nomination schemes, as well as their empirical computational complexity.

1. Introduction

Network data often exhibits underlying community structure, and there is a vast literature devoted to uncovering communities in complex networks; see, for example, Newman (2006), Von Luxburg (2007), Rohe et al. (2011) and Sussman et al. (2014). In many applications, one community in the network is of particular interest to the researcher. For example, in neuroscience connectomics, researchers might want to identify the region of the brain responsible for a particular

* Correspondence to: Department of Applied Math. and Stat., Johns Hopkins University, 3400 N Charles St, Baltimore, MD 21218, United States of America.
E-mail address: def@jhu.edu (D.E. Fishkind).

https://doi.org/10.1016/j.csda.2020.106916

neurological function; in a social network, a marketing company might want to find a group of users with similar interests; in an internet hyperlink network, a journalist might want to find blogs with a certain political leaning or subject matter. If we are given a few vertices known to be from the community of interest, and perhaps a few vertices known to not be from the community of interest, the task of *vertex nomination* is to order the remaining vertices in the network into a *nomination list*, with the aim of having a concentration of vertices from the community of interest at the top of the list; for alternate formulations of the vertex nomination problem, see Patsolic et al. (2017) and Lyzinski et al. (2019).

In Fishkind et al. (2015), three novel vertex nomination schemes were introduced: the canonical vertex nomination scheme $\mathcal{L}^C$, the likelihood maximization vertex nomination scheme $\mathcal{L}^{ML}$, and the spectral partitioning vertex nomination scheme $\mathcal{L}^P$. Under mild model assumptions, the canonical vertex nomination scheme $\mathcal{L}^C$ – which is the vertex nomination analogue of the Bayes' classifier – was proven to be the optimal vertex nomination scheme according to a mean average precision metric (see Definition 3). Unfortunately, $\mathcal{L}^C$ is not practical to implement on graphs with more than a few tens of vertices. The likelihood maximization vertex nomination scheme $\mathcal{L}^{ML}$ utilizes novel graph matching machinery, and is shown to be highly effective on both simulated and real data sets. However, $\mathcal{L}^{ML}$ is not practical to implement on graphs with more than a few thousand vertices. The spectral partitioning vertex nomination scheme $\mathcal{L}^P$ is less effective than the canonical and the likelihood maximization vertex nomination schemes on the small and moderately sized networks where the canonical and the likelihood maximization vertex nomination schemes can respectively be implemented in practice. Nonetheless, the spectral partitioning vertex nomination scheme has the significant advantage of being practical to implement on graphs with up to tens of millions of vertices.

1.1. Extending $\mathcal{L}^C$ and $\mathcal{L}^P$

In this paper we present extensions of the $\mathcal{L}^C$ and $\mathcal{L}^P$ vertex nomination schemes. Our extension of the canonical vertex nomination scheme $\mathcal{L}^C$, which we shall call the *canonical sampling vertex nomination scheme* and denote it as $\mathcal{L}^{CS}$, is an approximation of $\mathcal{L}^C$ that can be practically computed for graphs with hundreds of thousands of vertices, and our extension of the spectral partitioning vertex nomination scheme $\mathcal{L}^P$, which we shall call the *extended spectral partitioning vertex nomination scheme* and denote it as $\mathcal{L}^{EP}$, can be practically computed for graphs with close to one hundred thousand vertices, with significantly increased effectiveness (i.e. precision) over that of $\mathcal{L}^P$ when used on moderately sized networks.

While both $\mathcal{L}^{CS}$ and $\mathcal{L}^{EP}$ are practical to implement on very large graphs, the former has the important theoretical advantage of directly approximating the provably optimally precise vertex nomination scheme $\mathcal{L}^C$, with this approximation getting better and better when more and more sampling is used (and converging to $\mathcal{L}^C$ in this limit). However, as with $\mathcal{L}^C$, the canonical sampling scheme can be held back by the need to know/estimate the parameters of the underlying graph model before implementation. While this may be impractical in settings where these estimates are infeasible, $\mathcal{L}^{CS}$ allows us to approximately compute optimal precision in a larger array of synthetic models, thereby allowing us to better assess the performance of other, more feasibly implemented, procedures. Indeed, given unlimited computational resources (for sampling purposes), when the model parameters are known a priori or estimated to a suitable precision, $\mathcal{L}^{CS}$ would be more effective than every vertex nomination scheme other than $\mathcal{L}^C$.

In contrast, $\mathcal{L}^{EP}$ is implemented without needing to estimate the underlying graph model parameters; indeed, including known parameter estimates into the $\mathcal{L}^{EP}$ framework is nontrivial. This can lead to superior performance of $\mathcal{L}^{EP}$ versus $\mathcal{L}^{CS}$, especially in the setting where parameter estimates are necessarily highly variable. Additionally, given equal computational resources (i.e., when limiting the sampling allowed in $\mathcal{L}^{CS}$), $\mathcal{L}^{EP}$ is often more effective than $\mathcal{L}^{CS}$, even when the model parameters are well estimated.

See Fig. 1 for an informal visual representation that succinctly compares the various vertex nomination schemes on the basis of effectiveness (i.e. precision) and also computational practicality, as the scale of the number of vertices changes. The colors red, blue, green, purple, and orange correspond respectively to the canonical $\mathcal{L}^C$, canonical sampling $\mathcal{L}^{CS}$, likelihood maximization $\mathcal{L}^{ML}$, extended spectral partitioning $\mathcal{L}^{EP}$, and spectral partitioning $\mathcal{L}^P$ vertex nomination schemes. The lines dim to reflect increased computational burden. The red line on top represents the canonical vertex nomination scheme $\mathcal{L}^C$; it quickly dims out at a few tens of vertices, since at this point $\mathcal{L}^C$ is no longer practical to compute. Otherwise, the red line would have extended in a straight line across the figure, above all of the other lines, since it is the optimal nomination scheme (in the sense of precision), and is thus the benchmark for comparison of all of the other nomination schemes. Next, the dark/light/lighter blue regions correspond to the canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$; it is not a single line, but rather layers of lines for the different amounts of sampling that could be performed. As the number of vertices grows, $\mathcal{L}^{CS}$ requires more sampling – i.e. computational burden – to be more effective, hence the blue color lightens upwards in the figure, as it approaches the red line—or where the red line would have extended to. For graphs with few vertices, the dark blue line is just below the red line; indeed, the canonical sampling scheme is just as effective as the canonical scheme, and without much computational burden. Even with more vertices, with enough sampling we would have $\mathcal{L}^{CS}$ approaching $\mathcal{L}^C$, but with an ever increasing computational burden, hence the dimming of the blue towards the top of the figure. Next, the green line corresponds to the likelihood maximization vertex nomination scheme $\mathcal{L}^{ML}$; the green color dims out at a few thousand vertices, since at this point it is no longer practical to compute. Finally, the purple and orange lines, respectively, correspond to the extended spectral partitioning $\mathcal{L}^{EP}$, and spectral partitioning $\mathcal{L}^P$ vertex nomination schemes, the former being uniformly more effective then the latter. When there are only a few vertices the spectral methods are essentially useless, and these methods only become effective when there are a moderate number of

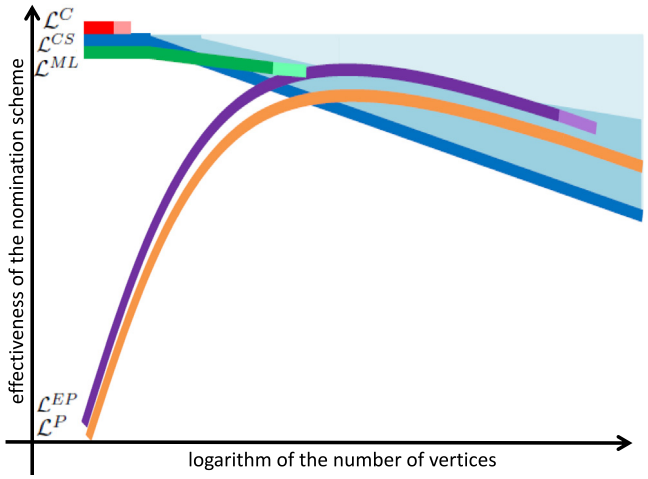


Fig. 1. A visual representation to summarize and compare the effectiveness (i.e. precision) and computational practicality of the vertex nomination schemes. This manuscript introduces the canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$ (blue) as an extension of the canonical vertex nomination scheme $\mathcal{L}^{C}$ (red), and introduces the extended spectral partitioning vertex nomination scheme $\mathcal{L}^{EP}$ (purple) as a refinement of the spectral partitioning vertex nomination scheme $\mathcal{L}^{P}$ (orange).

vertices. The extended spectral partitioning scheme is practical to compute until there are close to a hundred thousand vertices, while the spectral partitioning scheme is practical to compute even for many millions of vertices.

The paper is laid out as follows. In Section 3.1, we describe the canonical vertex nomination scheme, and prove its theoretical optimality in a slightly different model setting than considered in Fishkind et al. (2015). In Section 3.2, we use Markov chain Monte Carlo methods to extend the canonical vertex nomination scheme $\mathcal{L}^C$ to the canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$. In Section 3.3, we describe the spectral partitioning nomination scheme. In Section 3.4, we extend the spectral partitioning vertex nomination scheme $\mathcal{L}^P$ to the extended spectral partitioning vertex nomination scheme $\mathcal{L}^{EP}$, utilizing a more sophisticated clustering methodology than in $\mathcal{L}^P$, without an inordinately large sacrifice in scalability. In Section 4, we demonstrate and compare the performance of $\mathcal{L}^{EP}$ and $\mathcal{L}^{CS}$ on both simulated and real data sets.

2. Setting

We develop our vertex nomination schemes in the setting of the stochastic block model, a random graph model extensively used to model networks with underlying community structure. See, for example, Holland et al. (1983), Wang and Wong (1987) and Airoldi et al. (2008). The stochastic block model is a very simple random graph model that provides a principled approximation for more complicated network data (see, for example, Olhede and Wolfe, 2014; Wolfe and Olhede, 2013; Karrer and Newman, 2011), with the advantage that the theory associated with the stochastic block model is quite tractable.

The stochastic block model random graph is defined as follows; let K be a fixed positive integer.

Definition 1. A random graph $\mathbf{G}$ is an SBM$(K, \vec{n}, b, \Lambda)$ graph if

i. The vertex set V is the disjoint union of K sets $V = V_1 \sqcup V_2 \sqcup \cdots \sqcup V_K$ such that, for each $i = 1, 2, \ldots, K$, it holds that $|V_i| = n_i$. (For each i, V_i is called the ith **block**.)
ii. The **block membership function** $b : V \to \{1, 2, \ldots, K\}$ is such that, for all $v \in V$ and all $i = 1, 2, \ldots, K$, it holds that $b(v) = i$ if and only if $v \in V_i$.
iii. The **Bernoulli matrix** $\Lambda \in (0, 1)^{K \times K}$ is such that, for each pair of vertices $\{u, v\} \in \binom{V}{2}$, there is an edge between u and v (denoted $u \sim_{\mathbf{G}} v$) with probability $\Lambda_{b(u),b(v)}$, and the collection of indicator random variables $\{\mathbb{1}_{u \sim_{\mathbf{G}} v}\}_{\{u,v\} \in \binom{V}{2}}$ is independent.

In the setting of vertex nomination, we assume that b is only partially observed. Specifically, V is partitioned into two disjoint sets, S (the set of *seeds*) and A (the set of *ambiguous vertices*), and we assume that the values of b are known only on S. We denote the restriction of b to S as $b_{|S} : S \to \{1, 2, \ldots, K\}$. For each $i = 1, 2, \ldots, K$, we denote $A_i := V_i \cap A$, $S_i := V_i \cap S$, $m_i = |S_i|$, then we define $m := \sum_{i=1}^K m_i$, and $n := \sum_{i=1}^K n_i$. Of course, $|S| = m$ and $|A| = n - m$.

Given an SBM$(K, \vec{n}, b, \Lambda)$ model where the parameters are unknown, these parameters can be approximated in all of the usual ways utilizing a graph G realized from $\mathbf{G} \sim$SBM$(K, \vec{n}, b, \Lambda)$. First, K can be consistently estimated by spectral methods (such as in Fishkind et al., 2013; Wang and Bickel, 2017). Alternatively, since $b_{|S}$ is observed, we would

be observing K if we knew that $b_{\restriction S}$ was a surjective function. Given K, and assuming that the vertex memberships were realized via a multinomial distribution, then n_i can be estimated by $\frac{m_i}{m}n$, for each $i = 1, 2, \ldots, K$. Then, for any $i, j \in \{1, 2, \ldots, K\}$ such that $i \neq j$, we can estimate $\Lambda_{i,j}$ by the number of edges in the bipartite subgraph induced by S_i, S_j, divided by $m_i m_j$; i.e.,

$$\widehat{\Lambda}_{i,j} = \frac{|\{\{u, v\} \in E \text{ s.t. } u \in S_i,\ v \in S_j\}|}{m_i m_j}. \tag{1}$$

For $i = j$, we can estimate $\Lambda_{i,i}$ by the number of edges in the subgraph induced by S_i, divided by $\binom{m_i}{2}$; i.e.,

$$\widehat{\Lambda}_{i,i} = \frac{|\{\{u, v\} \in E \text{ s.t. } u, v \in S_i\}|}{\binom{m_i}{2}}. \tag{2}$$

In simulations, when it is useful or simplifying to do so, we assume that the model parameters K, $\vec{n}$, Λ are known. Else, they are estimated as above.

Next, the most general inference task here would be, given observed G from $\mathbf{G} \sim \text{SBM}(K, \vec{n}, b, \Lambda)$ and a partially observed block membership function $b_{\restriction S}$, to estimate the parameter b; that is, to estimate the remaining unobserved block memberships. Indeed, there are a host of graph clustering algorithms that could be used for this purpose; see, for example, (Rohe et al., 2011; Qin and Rohe, 2013; Sussman et al., 2012; Bickel et al., 2013; Newman, 2006; Von Luxburg, 2007) among others. However, in the vertex nomination (Marchette et al., 2011; Coppersmith and Priebe, 2012; Sun et al., 2012; Coppersmith, 2014; Fishkind et al., 2015) setting of this manuscript, the task of interest is much more specialized. We assume that there is only one block "of interest"—without loss of generality it is V_1—and we want to prioritize ambiguous vertices per the possibility of being from V_1. Specifically, our task is, given an observed G and a partially observed block membership function $b_{\restriction S}$, to order the ambiguous vertices A into a list such that there would be an abundance of vertices from V_1 that appear as near to the top of the list as can be achieved. More formally:

Definition 2. Given S, A, and $b_{\restriction S}$, a vertex nomination scheme $\mathcal{L}$ is a function $\mathcal{L} : \mathcal{G} \mapsto A!$ where $\mathcal{G}$ is the set of all graphs on vertex set $V = S \sqcup A$, and $A!$ is the set of all orderings of the set A. For any given $G \in \mathcal{G}$, denote the ordering $\mathcal{L}(G)$ of A as $(\mathcal{L}_{G,1}, \mathcal{L}_{G,2}, \ldots \mathcal{L}_{G,n-m})$; this ordering is also called the nomination list associated with $\mathcal{L}$ and G.

As in Fishkind et al. (2015), it is helpful for analysis to assume that for all graphs with symmetry (i.e., when a graph has a nontrivial automorphism group), that all vertex nomination schemes $\mathcal{L}$ assign such graphs to an empty nomination list. There is not much loss of generality in this, since the number of graphs with symmetry is very quickly negligible as the number of vertices increases (Erdos and Renyi, 1963; Polya, 1937). We also require that all vertex nomination schemes $\mathcal{L}$ have the following property: For any asymmetric $G, H \in \mathcal{G}$ such that G is isomorphic to H via isomorphism γ such that γ is the identity function on S, we require that $\gamma(\mathcal{L}_{G,i}) = \mathcal{L}_{H,i}$ for all i. In words, $\mathcal{L}$ should order the ambiguous vertices as if they are unlabeled.

The effectiveness of a vertex nomination scheme $\mathcal{L}$ is quantified in the following manner. Given a realization G of $\mathbf{G} \sim \text{SBM}(K, \vec{n}, b, \Lambda)$ and the partially observed block membership function b, and for any integer $j = 1, 2, \ldots, n - m$, define the *precision at depth j* of the list $\mathcal{L}(G)$ to be

$$\frac{|\{i \text{ such that } 1 \leq i \leq j,\ b(\mathcal{L}_{G,i}) = 1\}|}{j};$$

that is, the fraction of the first j vertices on the nomination list that are in the block of interest, V_1. The *average precision* of the list $\mathcal{L}(G)$ is defined to be the average of the precisions at depths $j = 1, 2, \ldots, n_1 - m_1$; that is, it is equal to

$$\frac{1}{n_1 - m_1} \sum_{j=1}^{n_1 - m_1} \frac{|\{i \text{ such that } 1 \leq i \leq j,\ b(\mathcal{L}_{G,i}) = 1\}|}{j}. \tag{3}$$

Of course, average precision is defined for a particular instantiation of G, and hence does not capture the behavior of $\mathcal{L}$ as G varies in the SBM model. To account for this, we define the *mean average precision*, the metric by which we will evaluate our vertex nomination schemes:

Definition 3. Let $\mathbf{G} \sim \text{SBM}(K, \vec{n}, b, \Lambda)$. The *mean average precision* of a vertex nomination scheme $\mathcal{L}$ is defined to be

$$MAP(\mathcal{L}) = \mathbb{E}\left(\frac{1}{n_1 - m_1} \sum_{j=1}^{n_1 - m_1} \frac{|\{i \text{ such that } 1 \leq i \leq j,\ b(\mathcal{L}_{\mathbf{G},i}) = 1\}|}{j} \right),$$

where the expectation is taken over the underlying probability space, the sample space being $\mathcal{G}$.

It is immediate that, for any given vertex nomination scheme $\mathcal{L}$, the mean average precision satisfies $MAP(\mathcal{L}) \in [0, 1]$, with values closer to 1 indicating a more successful nomination scheme; i.e., a higher concentration of vertices from V_1 near the top of the nomination list.

In the literature, mean average precision is often defined as the integral of the precision over recall. Herein, we focus on the definition of mean average precision provided in Definition 3 because, in the vertex nomination setting, recall is not as important as precision; the goal is explicitly to have an abundance of vertices of interest at the top of the list, and less explicitly about wanting all the vertices of interest to be high in the list.

3. Extending the vertex nomination schemes

In this section, we extend the canonical vertex nomination scheme $\mathcal{L}^C$ (described in Section 3.1) to a "sampling" version $\mathcal{L}^{CS}$ (defined in Section 3.2), and we extend the spectral partitioning vertex nomination scheme $\mathcal{L}^P$ (described in Section 3.3) to $\mathcal{L}^{EP}$ (defined in Section 3.4).

3.1. The canonical vertex nomination scheme $\mathcal{L}^C$

The canonical vertex nomination scheme $\mathcal{L}^C$, introduced in the paper Fishkind et al. (2015), is defined to be the vertex nomination scheme which orders the ambiguous vertices of A according to the order of their conditional probability – conditioned on G – of being members of the block of interest V_1. Indeed, it is intuitively clear why this would be an excellent (in fact, optimal) nomination scheme. However, since b is a parameter, this conditional probability is not yet meaningfully defined. We therefore expand the probability space of the SBM model given in Section 2, and construct a probability measure $\mathbb{Q}$ for which the canonical vertex nomination scheme $\mathcal{L}^C$ can be meaningfully defined, with its requisite conditional probabilities. The probability measure $\mathbb{Q}$ is constructed as follows:

Define Φ to be the collection of functions $\varphi : V \to \{1, 2, \ldots, K\}$ such that $\varphi(v) = b(v)$ for all $v \in S$, and such that $\left|\{v \in V : \varphi(v) = i\}\right| = n_i$ for all $i = 1, 2, \ldots, K$. Also, recall that $\mathcal{G}$ is the set of all graphs on V. The probability measure $\mathbb{Q}$ has sample space $\mathcal{G} \times \Phi$, and it is sampled from by first choosing $\varphi \in \Phi$ discrete-uniform randomly and then, conditioned on φ, G is chosen from the distribution SBM$(K, \vec{n}, \varphi, \Lambda)$. So, for all $G \in \mathcal{G}$, $\varphi \in \Phi$,

$$\mathbb{Q}(G, \varphi) = \frac{1}{\binom{n-m}{n_1-m_1, n_2-m_2, \ldots, n_K-m_K}} \prod_{i=1}^{K} \prod_{j=i}^{K} \left(\Lambda_{i,j}\right)^{e_{i,j}^{G,\varphi}} \left(1-\Lambda_{i,j}\right)^{c_{i,j}^{G,\varphi}}, \tag{4}$$

where $e_{i,j}^{G,\varphi}$ is defined as the number of edges in G such that φ of one endpoint is i and φ of the other endpoint is j, and we define $c_{i,j}^{G,\varphi} := n_i n_j - e_{i,j}^{G,\varphi}$ if $i \neq j$, and $c_{i,i}^{G,\varphi} := \binom{n_i}{2} - e_{i,i}^{G,\varphi}$. This probability measure, with uniform marginal distribution on Φ, reflects our situation where we have no prior knowledge of specific block membership for the ambiguous vertices (beyond block sizes). Note that $\mathbb{Q}$ is an intermediate measure used to show that $\mathcal{L}^C$ is optimal as stated in Theorem 4.

The first step in the canonical nomination scheme is to update this uniform distribution on Φ to reflect what is learned from the realization of the graph. Indeed, conditioning on any $G \in \mathcal{G}$, the conditional sample space of $\mathbb{Q}$ collapses to become Φ and, for any $\varphi \in \Phi$, we have by Bayes Rule that

$$\mathbb{Q}(\varphi|G) = \frac{\mathbb{Q}(G, \varphi)}{\sum_{\psi \in \Phi} \mathbb{Q}(G, \psi)} = \frac{\prod_{i=1}^{K} \prod_{j=i}^{K} \left(\Lambda_{i,j}\right)^{e_{i,j}^{G,\varphi}} \left(1-\Lambda_{i,j}\right)^{c_{i,j}^{G,\varphi}}}{\sum_{\psi \in \Phi} \prod_{i=1}^{K} \prod_{j=i}^{K} \left(\Lambda_{i,j}\right)^{e_{i,j}^{G,\psi}} \left(1-\Lambda_{i,j}\right)^{c_{i,j}^{G,\psi}}}. \tag{5}$$

In all that follows in this subsection, let $\mathbf{G}, \phi$ respectively denote the random graph and the random function, together distributed as $\mathbb{Q}$; in particular, the random $\mathbf{G}$ is $\mathcal{G}$-valued, and the random ϕ is Φ-valued. For each $v \in A$, the event $\phi(v) = 1$ is the event $\{\varphi \in \Phi : \varphi(v) = 1\}$ and, by Bayes' Rule,

$$\mathbb{Q}(\ \phi(v) = 1 \ \big|\ G\ ) = \frac{\sum_{\varphi \in \Phi : \varphi(v) = 1} \prod_{i=1}^{K} \prod_{j=i}^{K} \left(\Lambda_{i,j}\right)^{e_{i,j}^{G,\varphi}} \left(1-\Lambda_{i,j}\right)^{c_{i,j}^{G,\varphi}}}{\sum_{\varphi \in \Phi} \prod_{i=1}^{K} \prod_{j=i}^{K} \left(\Lambda_{i,j}\right)^{e_{i,j}^{G,\varphi}} \left(1-\Lambda_{i,j}\right)^{c_{i,j}^{G,\varphi}}}. \tag{6}$$

The canonical vertex nomination scheme $\mathcal{L}^C$ is then defined as ordering the vertices in A by decreasing value of $\mathbb{Q}(\phi(v) = 1|G)$ (with ties broken arbitrarily);

$$\begin{aligned}
&\mathcal{L}^C_{G,1} \in \operatorname{argmax}_{v \in A} \mathbb{Q}(\phi(v) = 1|G);\\
&\mathcal{L}^C_{G,2} \in \operatorname{argmax}_{v \in A \setminus \mathcal{L}^C_{G,1}} \mathbb{Q}(\phi(v) = 1|G);\\
&\qquad \vdots\\
&\mathcal{L}^C_{G,n-m} \in \operatorname{argmax}_{v \in A \setminus \left(\cup_{j=1}^{n-m-1} \mathcal{L}^C_{G,j}\right)} \mathbb{Q}(\phi(v) = 1|G).
\end{aligned} \tag{7}$$

In Fishkind et al. (2015) it is proved that the canonical vertex nomination scheme is an optimal vertex nomination scheme, in the sense of Theorem 4. We include the proof of Theorem 4 to reflect changes in our setting from the setting in Fishkind et al. (2015). Recall from the paragraph after Definition 2 that we assume that all vertex nomination schemes

assign graphs with symmetry to an empty nomination list. There is not much impact in this, since the number of graphs with symmetry is quickly negligible as the number of vertices increases (Erdos and Renyi, 1963; Polya, 1937). Then, for any asymmetric $G, H \in \mathcal{G}$ such that G is isomorphic to H via isomorphism γ such that γ is the identity function on S, we also required that $\gamma(\mathcal{L}_{G,i}) = \mathcal{L}_{H,i}$ for all i; in words, $\mathcal{L}$ should order the ambiguous vertices as if they are unlabeled. Clearly $\mathcal{L}^C$ satisfies this.

Theorem 4. *For any stochastic block model SBM($K, \vec{n}, b, \Lambda$) and vertex nomination scheme $\mathcal{L}$, it holds that $MAP(\mathcal{L}^C) \geq MAP(\mathcal{L})$.*

Proof of Theorem 4. For each $i = 1, 2, \ldots, n_1 - m_1$, define $\alpha_i := \frac{1}{n_1 - m_1} \sum_{j=i}^{n_1 - m_1} \frac{1}{j}$ and then, for each of $i = n_1 - m_1 + 1, n_1 - m_1 + 2, \ldots, n - m$, define $\alpha_i := 0$. Note that the sequence of α_i's is nonnegative and nonincreasing. Thus, for any other nonnegative and nonincreasing sequence of real numbers $a_1, a_2, \ldots, a_{n-m}$ and any rearrangement $a'_1, a'_2, \ldots, a'_{n-m}$ of the sequence $a_1, a_2, \ldots, a_{n-m}$, we have by the Rearrangement Inequality (Hardy et al., 1952) that

$$\sum_{i=1}^{n-m} \alpha_i a'_i \leq \sum_{i=1}^{n-m} \alpha_i a_i. \tag{8}$$

Next, consider any $\varphi, \varphi' \in \Phi$, and suppose that a function $\gamma : V \to V$ is bijective, that γ is the identity function on S, and that γ satisfies $\forall v \in A$, $\varphi(v) = \varphi'(\gamma(v))$. For any $G \in \mathcal{G}$, let $\gamma(G)$ denote the graph in $\mathcal{G}$ isomorphic to G via the isomorphism γ; it is clear that (under our assumptions, in particular suppose G is asymmetric) $\gamma(\mathcal{L}^C_{G,i}) = \mathcal{L}^C_{\gamma(G),i}$ for all i, since the canonical vertex nomination scheme orders the vertices as if they are unlabeled. Thus, since $\gamma : \mathcal{G} \to \mathcal{G}$ is clearly bijective, we have for all i that

$$\begin{aligned}
&\mathbb{Q}\left(\phi(\mathcal{L}^C_{\mathbf{G},i}) = 1 \,\middle|\, \phi = \varphi\right) = \sum_{v \in A \,:\, \varphi(v)=1} \mathbb{Q}\left(v = \mathcal{L}^C_{\mathbf{G},i} \,\middle|\, \phi = \varphi\right) \\
&= \frac{1}{\binom{n-m}{n_1-m_1, n_2-m_2, \ldots, n_K-m_K}} \sum_{v \in A \,:\, \varphi(v)=1} \sum_{G \in \mathcal{G} \,:\, v = \mathcal{L}^C_{G,i}} \mathbb{Q}(G, \varphi) \\
&= \frac{1}{\binom{n-m}{n_1-m_1, n_2-m_2, \ldots, n_K-m_K}} \sum_{v \in A \,:\, \varphi'(v)=1} \sum_{G \in \mathcal{G} \,:\, v = \mathcal{L}^C_{\gamma(G),i}} \mathbb{Q}(\gamma(G), \varphi') \\
&= \sum_{v \in A \,:\, \varphi'(v)=1} \mathbb{Q}\left(v = \mathcal{L}^C_{\mathbf{G},i} \,\middle|\, \phi = \varphi'\right) = \mathbb{Q}\left(\phi(\mathcal{L}^C_{\mathbf{G},i}) = 1 \,\middle|\, \phi = \varphi'\right);
\end{aligned}$$

since φ and φ' were arbitrary, the preceding is thus equal to (unconditioned) $\mathbb{Q}\left(\phi(\mathcal{L}^C_{\mathbf{G},i}) = 1\right)$. Hence, for all i, we have that

$$\mathbb{Q}\left(b(\mathcal{L}^C_{\mathbf{G},i}) = 1 \,\middle|\, \phi = b\right) = \mathbb{Q}\left(\phi(\mathcal{L}^C_{\mathbf{G},i}) = 1\right). \tag{9}$$

By the same reasoning, the vertex nomination scheme $\mathcal{L}$ also satisfies Eq. (9).

Now, to the main line of reasoning in the proof:

$$\begin{aligned}
MAP(\mathcal{L}^C) &= \mathbb{E}\left(\frac{1}{n_1 - m_1} \sum_{j=1}^{n_1-m_1} \frac{|\{i \text{ such that } 1 \leq i \leq j,\ b(\mathcal{L}^C_{\mathbf{G},i}) = 1\}|}{j} \,\middle|\, \phi = b\right) \\
&= \mathbb{E}\left(\sum_{i=1}^{n-m} \alpha_i \cdot \mathbb{1}\left[b(\mathcal{L}^C_{\mathbf{G},i}) = 1\right] \,\middle|\, \phi = b\right) \\
&= \sum_{i=1}^{n-m} \alpha_i \cdot \mathbb{Q}\left(b(\mathcal{L}^C_{\mathbf{G},i}) = 1 \,\middle|\, \phi = b\right) \\
&= \sum_{i=1}^{n-m} \alpha_i \cdot \mathbb{Q}\left(\phi(\mathcal{L}^C_{\mathbf{G},i}) = 1\right) \qquad \text{by Eq. (9)} \\
&= \sum_{i=1}^{n-m} \alpha_i \left(\sum_{G \in \mathcal{G}} \mathbb{Q}(G) \cdot \mathbb{Q}\left(\phi(\mathcal{L}^C_{G,i}) = 1 \,\middle|\, G\right)\right).
\end{aligned}$$

From this, we have

$$\begin{aligned}
MAP(\mathcal{L}^C) &= \sum_{G\in\mathcal{G}} \mathbb{Q}(G) \sum_{i=1}^{n-m} \alpha_i \cdot \mathbb{Q}\left(\phi(\mathcal{L}^C_{G,i}) = 1 \;\middle|\; G\right) \\
&\geq \sum_{G\in\mathcal{G}} \mathbb{Q}(G) \sum_{i=1}^{n-m} \alpha_i \cdot \mathbb{Q}\left(\phi(\mathcal{L}_{G,i}) = 1 \;\middle|\; G\right) \qquad \text{by definition of } \mathcal{L}^C, \text{ Eq. (8)} \\
&= \sum_{i=1}^{n-m} \alpha_i \cdot \mathbb{Q}\left(\phi(\mathcal{L}_{\mathbf{G},i}) = 1\right) \\
&= \sum_{i=1}^{n-m} \alpha_i \cdot \mathbb{Q}\left(b(\mathcal{L}_{\mathbf{G},i}) = 1 \;\middle|\; \phi = b\right) \\
&= \mathbb{E}\left(\frac{1}{n_1 - m_1} \sum_{j=1}^{n_1-m_1} \frac{|\{i \text{ such that } 1 \leq i \leq j,\ b(\mathcal{L}_{\mathbf{G},i}) = 1\}|}{j} \;\middle|\; \phi = b \right) \\
&= MAP(\mathcal{L}),
\end{aligned}$$

which completes the proof of Theorem 4. □

3.2. The canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$

The formula in Eq. (6) can be directly used to compute $\mathbb{Q}(\phi(v) = 1|G)$ for all $v \in A$, to obtain the ordering that defines the canonical vertex nomination scheme $\mathcal{L}^C$, but due to the burgeoning number of summands in the numerator and in the denominator of Eq. (6), this direct approach is computationally intractable, feasible only when the number of vertices is on the order of a few tens. We next introduce an extension of the canonical vertex nomination scheme called the *canonical sampling vertex nomination scheme* $\mathcal{L}^{CS}$. The purpose of the canonical sampling vertex nomination scheme is to provide a computationally tractable estimate $\widehat{\mathbb{Q}}(\phi(v) = 1|G)$ of $\mathbb{Q}(\phi(v) = 1|G)$, for all $v \in A$. The nomination list for $\mathcal{L}^{CS}$ consists of the vertices $v \in A$ ordered by nonincreasing values of $\widehat{\mathbb{Q}}(\phi(v) = 1|G)$, exactly as the nomination list for $\mathcal{L}^C$ consists of the vertices $v \in A$ ordered by nonincreasing values of $\mathbb{Q}(\phi(v) = 1|G)$.

Given the realized graph instance $G \in \mathcal{G}$ of the random graph $\mathbf{G}$, we obtain the approximation $\widehat{\mathbb{Q}}(\phi(v) = 1|G)$ of $\mathbb{Q}(\phi(v) = 1|G)$ for all $v \in A$ by sampling from the conditioned-on-G probability space $\mathbb{Q}(\cdot|G)$ on Φ, then, for each $v \in A$, $\widehat{\mathbb{Q}}(\phi(v) = 1|G)$ is defined as the fraction of the sampled functions (Φ is a set of functions) that map v to the integer 1. The formula for the conditional probability distribution $\mathbb{Q}(\cdot|G)$ is given in Eq. (5); unfortunately, straightforward sampling from this distribution is hampered by the intractability of directly computing the denominator of Eq. (5). Fortunately, sampling in this setting can be achieved via Metropolis–Hastings Markov chain Monte Carlo. For relevant background on Markov chain Monte Carlo, see, for example, Gelman et al. (2014, Chapter 11) or Aldous and Fill (2002, Chapter 11).

The base chain that we employ in our Markov chain Monte Carlo approach is the well-studied Bernoulli–Laplace diffusion model (Feller, 2008). The state space for the Markov chain is the set Φ, and the one-step transition probabilities, denoted $\mathrm{P}(\cdot,\cdot)$, for this chain are defined, for all $\varphi, \varphi' \in \Phi$, as

$$\mathrm{P}(\varphi, \varphi') = \frac{\mathbb{1}\{d(\varphi, \varphi') = 2\}}{\binom{n-m}{2} - \sum_{i=1}^{K} \binom{n_i - m_i}{2}},$$

where $d(\varphi, \varphi') := |\{v \text{ such that } \varphi(v) \neq \varphi'(v)\}|$. In other words, if at state φ, a move transpires as follows. A pair of vertices $\{u, v\} \in \binom{A}{2}$ is chosen discrete-uniformly at random, conditional on the fact that $\varphi(u) \neq \varphi(v)$, and then the move is to state φ', which is defined as agreeing with φ, except that $\varphi'(u)$ and $\varphi'(v)$ are defined respectively as $\varphi(v)$ and $\varphi(u)$. We will see shortly that the simplicity of this base chain greatly simplifies the computations needed to employ Metropolis–Hastings with target distribution $\mathbb{Q}(\cdot|G)$.

The Metropolis–Hastings chain has state space Φ, and one-step transition probabilities, $\widehat{\mathrm{P}}(\cdot,\cdot)$ defined for all $\varphi, \varphi' \in \Phi$ as

$$\widehat{\mathrm{P}}(\varphi, \varphi') = \frac{\mathbb{1}\{d(\varphi, \varphi') = 2\}}{\binom{n-m}{2} - \sum_{i=1}^{K} \binom{n_i - m_i}{2}} \min\left\{ 1,\ \frac{\prod_{i=1}^{K} \prod_{j=i}^{K} \left(\Lambda_{i,j}\right)^{e_{i,j}^{G,\varphi'}} \left(1 - \Lambda_{i,j}\right)^{c_{i,j}^{G,\varphi'}}}{\prod_{i=1}^{K} \prod_{j=i}^{K} \left(\Lambda_{i,j}\right)^{e_{i,j}^{G,\varphi}} \left(1 - \Lambda_{i,j}\right)^{c_{i,j}^{G,\varphi}}} \right\} \text{ if } \varphi \neq \varphi';$$

$$\widehat{\mathrm{P}}(\varphi, \varphi) = 1 - \sum_{\varphi'' \in \Phi : \varphi'' \neq \varphi} \widehat{\mathrm{P}}(\varphi, \varphi'').$$

In other words, if at state φ, a candidate state φ' is proposed according to $\mathrm{P}(\varphi, \cdot)$ and is independently accepted as the next state of the Markov chain with probability $\min\left\{1, \frac{\mathbb{Q}(\varphi'|G)}{\mathbb{Q}(\varphi|G)}\right\}$. It is immediate that the stationary distribution for $\widehat{\mathrm{P}}$ is $\mathbb{Q}(\cdot|G)$ and that the chain is reversible with respect to $\mathbb{Q}(\cdot|G)$; that is, for any $\varphi, \varphi' \in \Phi$, $\mathbb{Q}(\varphi|G) \cdot \widehat{\mathrm{P}}(\varphi, \varphi') = \mathbb{Q}(\varphi'|G) \cdot \widehat{\mathrm{P}}(\varphi', \varphi)$.

Note that the simplicity of the underlying base chain greatly aids in the speedy computation of $\frac{\mathbb{Q}(\varphi'|G)}{\mathbb{Q}(\varphi|G)}$ during the computation of transition probabilities $\widehat{\mathrm{P}}$. Indeed, since φ and φ' for which we might want to compute $\widehat{\mathrm{P}}(\varphi, \varphi')$ are such that $d(\varphi, \varphi') = 2$, we would have that φ and φ' differ only on two vertices, call them u, v, and say that i and j are such that $\varphi(u) = i$ and $\varphi(v) = j$. Then

$$\frac{\mathbb{Q}(\varphi'|G)}{\mathbb{Q}(\varphi|G)} = \prod_{w \in V : w \neq u, w \neq v} \left(\frac{\Lambda_{\varphi(w),j}(1 - \Lambda_{\varphi(w),i})}{\Lambda_{\varphi(w),i}(1 - \Lambda_{\varphi(w),j})} \right)^{\left\{ \begin{smallmatrix} 1 & \text{if } w \sim_G u, w \not\sim_G v \\ -1 & \text{if } w \not\sim_G u, w \sim_G v \\ 0 & \text{else} \end{smallmatrix} \right\}}. \tag{10}$$

This reduces the number of operations to compute $\frac{\mathbb{Q}(\varphi'|G)}{\mathbb{Q}(\varphi|G)}$ from $O(n^2)$ down to $O(n)$. As an implementation note, in practice we would utilize a logarithm to convert Eq. (10) from a multiplicative expression into an additive expression, which will greatly reduce round off error that can arise when working with numbers that are orders of magnitude different from each other.

Now, the canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$ is defined in the exact same manner as $\mathcal{L}^{C}$, except that, for all $v \in A$, the value $\widehat{\mathbb{Q}}(\phi(v) = 1|G)$ is approximated as follows. Denoting the Metropolis–Hastings Markov chain by $(X_t)_{t=0}^{\infty}$, we set $X_0 \sim$Uniform(Φ). After evolving the chain past a "burn-in" period, T, we approximate $\mathbb{Q}(\phi(v) = 1|G)$ via

$$\widehat{\mathbb{Q}}(\phi(v) = 1|G) = \frac{|\{s \text{ such that } T < s \leq T + t, \text{ and } X_s(v) = 1\}|}{t},$$

for a predetermined number of Metropolis–Hastings steps t. For fixed T, we then have as an immediate consequence of the Ergodic Theorem (see, for example, Aldous and Fill, 2002, Chp. 2, Thm. 1) that $\lim_{t\to\infty} \widehat{\mathbb{Q}}(\phi(v) = 1|G) = \mathbb{Q}(\phi(v) = 1|G)$ for each $v \in A$ (indeed, our Metropolis–Hastings chain is aperiodic, recurrent and finite state).

In this paper, we do not address how to choose a suitable burn-in T for a given implementation of $\mathcal{L}^{CS}$, instead focusing on a feasible burn-in given limited computational resources. Practically, there are a bevy of methods for approximating T, see for example those in Gelman et al. (2014) and Gilks et al. (1995). Regarding mixing time, there is an unfortunate dearth of rigorous mixing time computations for general, non-unimodal Metropolis–Hastings algorithms (see the discussion in Diaconis and Saloff-Coste, 1998; Johndrow and Smith, 2018), and such analysis is beyond the scope of this paper. Our choice of the Bernoulli–Laplace base chain is for its fast and efficient implementation of the sampling procedure, although we have no guarantee or expectation of optimal mixing time.

3.3. The spectral partitioning vertex nomination scheme

We now review the spectral partitioning vertex nomination scheme $\mathcal{L}^{P}$ from Fishkind et al. (2015); afterwards, in Section 3.4, $\mathcal{L}^{P}$ will be extended to the vertex nomination scheme $\mathcal{L}^{EP}$.

As in Section 2, we assume here that the graph G is realized from an SBM$(K, \vec{n}, b, \Lambda)$ distribution, where K is known. Furthermore, we assume that the values of the block membership function b are known only on the set of seeds S, and are not known on the set of ambiguous vertices $A = V \backslash S$. In contrast to Section 3.1, here we do not need to assume that $\vec{n}$ and Λ are explicitly known or estimated, except that $d := \text{rank } \Lambda$ is known, or an upper bound for d is known. As before, say that V_1 is the "block of interest".

The spectral partitioning vertex nomination scheme $\mathcal{L}^{P}$ is computed in three stages; first is the adjacency spectral embedding of G, then clustering of the embedded points, and then ranking the ambiguous vertices into the nomination list. (The first two of these stages are collectively called *adjacency spectral clustering*; for a good reference, see Von Luxburg, 2007.) We begin by describing the first stage, adjacency spectral embedding:

Definition 5. Let graph G have adjacency matrix $\mathcal{A}$, and suppose $(\mathcal{A}^{\top}\mathcal{A})^{1/2}$ has eigendecompostion

$$(\mathcal{A}^{\top}\mathcal{A})^{1/2} = \left[U|\tilde{U}\right]\left[D \oplus \tilde{D}\right]\left[U|\tilde{U}\right]^{\top};$$

i.e., $U \in \mathbb{R}^{n\times d}$, $\left[U|\tilde{U}\right] \in \mathbb{R}^{n\times n}$ is orthogonal, $\left[D \oplus \tilde{D}\right] \in \mathbb{R}^{n\times n}$ is diagonal, and the diagonal of $D \in R^{d\times d}$ is composed of the d greatest eigenvalues of $(\mathcal{A}^{\top}\mathcal{A})^{1/2}$ in nonincreasing order. The d-dimensional *adjacency spectral embedding* of G is then given by $\widehat{X} = UD^{1/2}$. In particular, for each $v \in V$, the row of $\widehat{X}$ corresponding to v, denoted $\widehat{X}_v$, is the embedding of v into $\mathbb{R}^d$.

After the adjacency spectral embedding, the second stage is to cluster the embedded vertices – i.e. the associated points in $\mathbb{R}^d$ – using the k-means clustering algorithm (MacQueen, 1967). The clusters so obtained are estimates of the different blocks, and the cluster containing the most vertices from $S_1 := S \cap V_1$ is an estimate of the block of interest V_1; let c denote the centroid of this cluster. (Note that this clustering step, as described here for $\mathcal{L}^{P}$, is fully unsupervised, not taking advantage of the observed memberships of the vertices in S. In Section 3.4, incorporating these labels into a semi-supervised clustering step is a natural way to extend $\mathcal{L}^{P}$ and improve performance.)

The third stage is ranking the ambiguous vertices into the nomination list; the vertices are nominated based on their Euclidean distance from c, the centroid of the cluster which is the estimate for the block of interest. Specifically, define:

$$\begin{aligned}
\mathcal{L}^P_{G,1} &\in \operatorname{argin}_{v\in A}\|v-c\|_2;\\
\mathcal{L}^P_{G,2} &\in \operatorname{argmin}_{v\in A\setminus\mathcal{L}^P_{G,1}}\|v-c\|_2;\\
&\vdots\\
\mathcal{L}^P_{G,n-m} &\in \operatorname{argmin}_{v\in A\setminus\left(\cup_{j=1}^{n-m-1}\mathcal{L}^P_{G,j}\right)}\|v-c\|_2.
\end{aligned} \tag{11}$$

For definiteness, any ties in the above procedure should be broken by choosing uniform-randomly from the choices. This concludes the definition of the spectral partitioning vertex nomination scheme $\mathcal{L}^P$.

Under mild assumptions, it is proven in Lyzinski et al. (2014) that, in the limit, adjacency spectral partitioning almost surely perfectly clusters the vertices of G into the true blocks. This fact was leveraged in Fishkind et al. (2015) to prove that if $m_1 > 0$ and there exists a $\gamma > 0$ such that for all $i = 1, 2, \ldots, K$, $n_i \geq \gamma \cdot n^{3/4+\gamma}$, then $\lim_{n\to\infty}\mathrm{MAP}(\mathcal{L}^P) = 1$.

If d is unknown, singular value thresholding (Chatterjee, 2014) can be used to estimate d from a partial SCREE plot (Zhu and Ghodsi, 2006). We note that the results of Fishkind et al. (2013) suggest that there will be little performance lost if d is moderately overestimated. Additionally, if K is unknown then it can be estimated by optimizing the silhouette width of the resulting clustering (Kaufman and Rousseeuw, 2009). A key advantage of the spectral nomination scheme is that, unlike $\mathcal{L}^C$, Λ and $\vec{n}$ need not be estimated before applying the scheme.

3.4. The extended spectral partitioning vertex nomination scheme

In this section, we extend the spectral partitioning vertex nomination scheme $\mathcal{L}^P$ (described in the previous section) to the extended spectral partitioning vertex nomination scheme $\mathcal{L}^{EP}$. Just like in computing $\mathcal{L}^P$, computing the extended spectral partitioning vertex nomination scheme $\mathcal{L}^{EP}$ starts with adjacency spectral embedding. Whereas the next stage of $\mathcal{L}^P$ is unsupervised clustering using the k-means algorithm, $\mathcal{L}^{EP}$ will instead utilize a semi-supervised clustering procedure which we describe below.

There are numerous ways to incorporate the known block memberships for S into the clustering step of adjacency spectral clustering (see, for example, Wagstaff et al., 2001; Yoder and Priebe, 2014). The results of Athreya et al. (2015) suggest that, for each vertex v of $\mathbf{G}$, the distribution of v's embedding $\widehat{X}_v \in \mathbb{R}^d$ is approximately normal, with parameters that depend only on which block v is a member of, and this normal approximation gets closer to exact as n grows. We thus model $\mathbf{G}$'s embedded vertices as independent draws from a K-component Gaussian mixture model (except for vertices of S, where the Gaussian component is specified); i.e., there exists a fixed nonnegative vector $\pi := (\pi_1, \pi_2, \ldots, \pi_K) \in \mathbb{R}^K$ satisfying $\sum_{k=1}^K \pi_k = 1$, and for each $k = 1, 2, \ldots, K$, there exists $\mu^{(k)} \in \mathbb{R}^d$ and $\Sigma^{(k)} \in \mathbb{R}^{d\times d}$ such that, independently for each vertex $v \in A$, the block of v is $1, 2, \ldots, K$ with respective probabilities $\pi_1, \pi_2, \ldots, \pi_K$, and then, conditioning on model block membership – say the block of v is k – the distribution of $\widehat{X}_v$ is Normal$(\mu^{(k)}, \Sigma^{(k)})$. If $\boldsymbol{\mu}$ denotes the sequence of mean vectors $(\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(K)})$, $\boldsymbol{\sigma}$ denotes the sequence of covariance matrices $(\Sigma^{(1)}, \Sigma^{(2)}, \ldots, \Sigma^{(K)})$, and (random) $\varphi : V \to \{1, 2, \ldots, K\}$ denotes the Gaussian mixture model block membership function – i.e., for each $v \in V$ and $k \in \{1, 2, \ldots, K\}$, it holds that $\varphi(v) = k$ precisely when the Gaussian mixture model places v in block k – then the complete data log-likelihood function can be written as

$$\ell(\pi, \boldsymbol{\mu}, \boldsymbol{\sigma})_{\widehat{X},\varphi} = \sum_{k=1}^K \sum_{v\in S_k} \log\left(f_{\mu^{(k)},\Sigma^{(k)}}(\widehat{X}_v)\right) + \sum_{k=1}^K \sum_{v\in A} \mathbb{1}_{\varphi(v)=k} \log\left(\pi_k f_{\mu^{(k)},\Sigma^{(k)}}(\widehat{X}_v)\right), \tag{12}$$

which meaningfully incorporates the seeding information contained in S.

If $\vec{n}$ is known (indeed, it was assumed to be known in the formulation of $\mathcal{L}^C$, but was not assumed to be known in the formulation of $\mathcal{L}^P$) then, for each $k = 1, 2, \ldots, K$, we would substitute $\frac{n_k}{n}$ in place of π_k.

With this model is place, it is natural to cluster the rows of $\widehat{X}$ using a (semi-supervised) Gaussian mixture model (GMM) clustering algorithm rather than (unsupervised) k-means employed by $\mathcal{L}^P$. We now return to the description of the extended spectral partitioning vertex nomination scheme $\mathcal{L}^{EP}$ after the first stage – adjacency spectral embedding – has been performed. The next stage – clustering – can be cast as the problem of uncovering the latent $\mathbb{1}_{\varphi(v)=k}$'s as are present in the log-likelihood in Eq. (12). We employ a semi-supervised modification of the model-based `Mclust` Gaussian mixture model methodology of Fraley and Raftery (2002, 2006); we call this modification `ssMclust`; note that `ssMclust` first appeared in Yoder and Priebe (2014), and we include a brief outline of its implementation below for the sake of completeness.

As in Fraley and Raftery (2002), `ssMclust` uses the expectation–maximization (EM) algorithm to approximately find the maximum likelihood estimates of Eq. (12), denote them by $\hat{\pi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}$. For each $v \in A$, the cluster of $\widehat{X}_v$—which is an

Table 1
List of the `ssMclust` covariance parameterizations we consider. In the above, I is the identity matrix; D's are diagonal matrices; and U's represent matrices of orthonormal eigenvectors. If "k" is a subscript on any symbol then that parameter is allowed to vary across clusters and, if not, then the parameter must remain fixed across clusters. This table is expanded from Table 1 in Fraley and Raftery (2006).

Name	Applicable to	$\Sigma^{(k)}$	Volume	Shape	Orientation
E	$\mathbb{R}$	λ	Equal	NA	NA
V	$\mathbb{R}$	λ_k	Varying	NA	NA
X	$\mathbb{R}, K=1$	λ	NA	NA	NA
EII	$\mathbb{R}^d$	λI	Equal	Equal, spherical	Coordinate axes
VII	$\mathbb{R}^d$	$\lambda_k I$	Varying	Equal, spherical	Coordinate axes
EEI	$\mathbb{R}^d$	λD	Equal	Equal, ellipsoidal	Coordinate axes
VEI	$\mathbb{R}^d$	$\lambda_k D$	Varying	Equal, ellipsoidal	Coordinate axes
EVI	$\mathbb{R}^d$	λD_k	Equal	Varying, ellipsoidal	Coordinate axes
VVI	$\mathbb{R}^d$	$\lambda_k D_k$	Varying	Varying, ellipsoidal	Coordinate axes
EEE	$\mathbb{R}^d$	λUDU^T	Equal	Equal, ellipsoidal	Equal
EVE	$\mathbb{R}^d$	λUD_kU^T	Equal	Varying, ellipsoidal	Equal
VEE	$\mathbb{R}^d$	$\lambda_k UDU^T$	Varying	Equal, ellipsoidal	Equal
VVE	$\mathbb{R}^d$	$\lambda_k UD_kU^T$	Varying	Varying, ellipsoidal	Equal
EEV	$\mathbb{R}^d$	$\lambda U_kDU_k^T$	Equal	Equal, ellipsoidal	Varying
VEV	$\mathbb{R}^d$	$\lambda_k U_kDU_k^T$	Varying	Equal, ellipsoidal	Varying
EVV	$\mathbb{R}^d$	$\lambda U_kD_kU_k^T$	Equal	Varying, ellipsoidal	Varying
VVV	$\mathbb{R}^d$	$\lambda_k U_kD_kU_k^T$	Varying	Varying, ellipsoidal	Varying
XII	$\mathbb{R}^d, K=1$	λI	NA	Spherical	Coordinate Axes
XXI	$\mathbb{R}^d, K=1$	λD	NA	Ellipsoidal	Coordinate Axes
XXX	$\mathbb{R}^d, K=1$	λUDU^T	NA	Ellipsoidal	NA

estimate for the block of v—is then set to be

$$\widehat{\varphi}(v) := \text{argmax}_{k\in\{1,2,\ldots,K\}}\hat{\pi}_k f_{\hat{\mu}^{(k)},\widehat{\Sigma}^{(k)}}(\widehat{X}_v).$$

Details of the implementation of the semi-supervised EM algorithm can be found in McLachlan and Peel (2004), Yoder (2016) and Yoder and Priebe (2014), and are omitted here for brevity. We note here that we initialize the class assignments in the EM algorithm by first running the semi-supervised k-means++ algorithm of Yoder and Priebe (2017) on $\widehat{X}$. This initialization, in practice, has the effect of greatly reducing the running time of the EM step in `ssMclust`; see Yoder (2016).

Like in `Mclust`, the `ssMclust` framework balances model fit versus model parsimony. Like in `Mclust`, we use the Bayesian Information Criterion (BIC) to assess the quality of the clustering given by the Gaussian Mixture Models with density structure $f_{\widehat{X}_v} = \sum_{k=1}^{K}\pi_k f_{\mu^{(k)},\Sigma^{(k)}}$ over a range of K and various Gaussian parameterizations. The geometry of the kth cluster is determined by the structure of Σ_k; see Table 1 for a comprehensive list of the covariance structures we consider in `ssMclust`. While the more complicated geometric structure allows for a better fit of the data, this comes at the price of model complexity; i.e., more parameters to estimate.

The BIC penalty employed in `Mclust` and `ssMclust` rewards model fit, and it penalizes model complexity. Given model M, the BIC is usually defined as

$$\text{BIC}(M) = 2\max_{(\pi,\boldsymbol{\mu},\boldsymbol{\sigma})\in M}\ell(\pi,\boldsymbol{\mu},\boldsymbol{\sigma})_{\widehat{X},\varphi} - \tau_M\log n,$$

where $\max_{(\pi,\boldsymbol{\mu},\boldsymbol{\sigma})\in M}\ell(\pi,\boldsymbol{\mu},\boldsymbol{\sigma})_{\widehat{X},\varphi}$ is the maximized log-likelihood in Eq. (12), τ_M is the number of parameters estimated in model M (i.e., the number of parameters in $(\pi,\boldsymbol{\mu},\boldsymbol{\sigma})$ that need to be estimated), and n the number of observed data points. In the present semi-supervised setting, we propose an adjusted BIC that only penalizes the model complexity of the unsupervised data points, namely

$$\text{BIC}'(M) = 2\max_{(\pi,\boldsymbol{\mu},\boldsymbol{\sigma})\in M}\ell(\pi,\boldsymbol{\mu},\boldsymbol{\sigma})_{\widehat{X},\varphi} - \tau_M\log(n-m). \tag{13}$$

If $\lim_{n\to\infty} m/n = 0$, then $|\text{BIC}(M) - \text{BIC}'(M)| = o(1)$, but even in this setting, empirical evidence suggests the less parsimonious models allowed by $\text{BIC}'(M)$ provide a better model fit than the more parsimonious $\text{BIC}(M)$. Intuitively, the complexity introduced by the largely constrained supervised datum should be lower than that of the unconstrained unsupervised datum, which is reflected in the modified $\text{BIC}'(M)$; see Yoder (2016).

The `ssMclust` algorithm proceeds by maximizing the log-likelihood via the EM algorithm over a range of models $M\in\mathcal{M}$, and then uses the BIC penalty (13) to select the best fitting model, defined via

$$\widehat{M} = \text{argmax}_{M\in\mathcal{M}}\text{BIC}'(M).$$

Algorithm 1: Extended Spectral Partitioning Vertex Nomination Scheme

Input: Graph G on vertices $S \cup A$ (seeds, ambiguous); $n := |S \cup A|$, $m := |S|$
$b\restriction_S$ (block assignments of seeds)
d (embedding dimension)
$\mathcal{K}$ (maximum number of clusters to consider)
$\mathcal{M}$ (set of models to consider)
Output: $\mathcal{L}^{EP}$ (nomination scheme)

$\widehat{X} \leftarrow$ adjacency spectral embedding of G into $\mathbb{R}^d$;
foreach $M \in \mathcal{M}$ **do**
  Initialize the class labels using the semi-supervised $K_M - means++$ algorithm;
  $\ell_M \leftarrow$ max of complete log-likelihood under model M computed via the EM algorithm;
  $\text{BIC}'(M) \leftarrow 2\ell_M - \tau_M \log(n-m)$, where τ_M is the number of parameters estimated in M;
$\widehat{M} \leftarrow \text{argmax}_{M\in\mathcal{M}}\text{BIC}'(M)$.
$\mathcal{L}^{EP} \leftarrow$ nomination of the vertices of A according to Eq. (14) under model $\widehat{M}$
return $\mathcal{L}^{EP}$

Slightly abusing notation, let $(\hat{\pi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}) := (\hat{\pi}_{\widehat{M}}, \hat{\boldsymbol{\mu}}_{\widehat{M}}, \hat{\boldsymbol{\sigma}}_{\widehat{M}})$ be maximum likelihood estimates of $(\pi, \boldsymbol{\mu}, \boldsymbol{\sigma})$ in model $\widehat{M}$. The $\mathcal{L}^{EP}$ scheme then nominates the vertices in A via

$$\begin{aligned}
\mathcal{L}^{EP}_{G,1} &\in \text{argmax}_{v\in A}\hat{\pi}_1 f_{\hat{\mu}^{(1)},\hat{\Sigma}^{(1)}}(\widehat{X}_v);\\
\mathcal{L}^{EP}_{G,2} &\in \text{argmax}_{v\in A\setminus\mathcal{L}^{EP}_{G,1}}\hat{\pi}_1 f_{\hat{\mu}^{(1)},\hat{\Sigma}^{(1)}}(\widehat{X}_v);\\
&\vdots\\
\mathcal{L}^{EP}_{G,n-m} &\in \text{argmax}_{v\in A\setminus\left(\cup_{j=1}^{n-m-1}\mathcal{L}^{EP}_{G,j}\right)}\hat{\pi}_1 f_{\hat{\mu}^{(1)},\hat{\Sigma}^{(1)}}(\widehat{X}_v).
\end{aligned} \tag{14}$$

Details of the $\mathcal{L}^{EP}$ scheme are summarized in Algorithm 1.

In the case of a *quasi-seeding* – where b is observed for vertices in S_1 but for vertices in $S \setminus S_1$ it is only observed that the vertices are not in V_1 – the complete data log-likelihood becomes

$$\begin{aligned}
\ell(\pi, \boldsymbol{\mu}, \boldsymbol{\sigma})_{\widehat{X},\varphi} = &\sum_{v\in S_1}\log\left(f_{\mu^{(1)},\Sigma^{(1)}}(\widehat{X}_v)\right) + \sum_{k=2}^{K}\sum_{v\in S\setminus S_1}\mathbb{1}_{\varphi(v)=k}\log\left(\frac{\pi_k}{1-\pi_1}f_{\mu^{(k)},\Sigma^{(k)}}(\widehat{X}_v)\right)\\
&+\sum_{k=1}^{K}\sum_{v\in A}\mathbb{1}_{\varphi(v)=k}\log\left(\pi_k f_{\mu^{(k)},\Sigma^{(k)}}(\widehat{X}_v)\right),
\end{aligned}$$

and Algorithm 1 can be applied with this log-likelihood in place of Eq. (12). The ability of the `ssMclust` algorithm to seamlessly handle this scenario is a major advantage over other semi-supervised clustering techniques (e.g., logistic regression, random forest, etc.).

4. Experimental results

In this section, we demonstrate the effectiveness (in the sense of precision) and scalability of our vertex nomination schemes, the canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$ and the extended spectral partitioning vertex nomination scheme $\mathcal{L}^{EP}$, on both real and synthetic data. As mentioned in Section 1, the canonical vertex nomination scheme $\mathcal{L}^{C}$ is optimally effective (in the sense of precision) but does not scale, and the spectral partitioning vertex nomination scheme $\mathcal{L}^{P}$ scales well but is not nearly as effective as $\mathcal{L}^{C}$ on small to medium scale networks. (Indeed, $\mathcal{L}^{P}$ obtains nearly chance performance on small graphs). We illustrate in this section that $\mathcal{L}^{CS}$ and $\mathcal{L}^{EP}$ both scale and are very effective at multiple scales, markedly improving over their forerunners.

Each example in this section consists of *nMC* Monte Carlo replicates, for some preselected positive integer *nMC*; that is, we obtain *nMC* realizations of the underlying experiment, thus obtaining *nMC* nomination lists—for each of the vertex nomination schemes that are compared. For each vertex nomination scheme, the mean (average) of the *nMC* average precisions obtained will be referred to as the *empirical mean average precision* under the vertex nomination scheme. For each vertex nomination scheme and each nomination list position i, the fraction of the *nMC* nomination lists in which the ith list-position (vertex) was truly in V_1 is the *empirical probability that nomination list position i is in V_1* under the vertex nomination scheme. All of the figures in this section consist of plotting the empirical probabilities of nomination lists' position being in V_1 (on the y-axis) against the respective position in the nomination list (on the x-axis).

Table 2
Experimental parameters for the stochastic block model simulations.

Scale of experiment		$\vec{m}$	$\vec{n}-\vec{m}$	|A|	α
Small scale	small–small-scale	[4, 0, 0]	[4, 3, 3]	10	1
	medium–small-scale	[4, 0, 0]	[7, 4, 4]	15	1
	large–small-scale	[4, 0, 0]	[8, 5, 4]	17	1
Medium scale		[20, 0, 0]	[200, 150, 150]	500	0.3
Large scale		[40, 0, 0]	[4000, 3000, 3000]	10000	0.13

Table 3
Small scale experiment. Comparing $\mathcal{L}^{C}$ and $\mathcal{L}^{CS}$ by average runtime, empirical MAP.

Scale of experiment	$\|A\|$	Avg. running time (in s)		MAP	
		$\mathcal{L}^{C}$	$\mathcal{L}^{CS}$	$\mathcal{L}^{C}$	$\mathcal{L}^{CS}$
small–small-scale	10	1.12	.0335	.6934	.6901
medium–small-scale	15	128	.0453	.7632	.7530
large–small-scale	17	871	.0489	.8182	.8086

Note that we distinguish *nMC*, defined above, from *nMCMC*, which will denote the number of Markov chain Monte Carlo steps used in computing $\mathcal{L}^{CS}$; unless otherwise specified, we use $nMCMC/2$ steps for burn-in, and the other $nMCMC/2$ steps for actual sampling.

4.1. Simulation experiments

In this subsection, Section 4.1, we perform simulation experiments for a stochastic block model at three scales: the underlying model used here is $G \sim \text{SBM}\left(3, \vec{n}, b, \Lambda_{\alpha}\right)$ where

$$\Lambda_{\alpha} := \alpha \begin{bmatrix} 0.5 & 0.3 & 0.4 \\ 0.3 & 0.8 & 0.6 \\ 0.4 & 0.6 & 0.3 \end{bmatrix} + (1-\alpha) \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix},$$

for $\alpha \in [0, 1]$. We consider three experimental scales, summarized below in Table 2.

The parameter α allows us to control how stochastically differentiated the blocks are from one another; indeed, as α decreases the blocks become more stochastically homogeneous and, when $\alpha = 0$, there is effectively only one block (the graph is Erdős–Rényi). Note that the block of interest, V_1, is of intermediate density; less densely intraconnected than V_2 and more than V_3. The true model parameters—$K, \vec{n}, \Lambda$—are used when implementing $\mathcal{L}^{C}$, $\mathcal{L}^{CS}$, (as well as $\mathcal{L}^{ML}$—the likelihood maximization vertex nomination scheme introduced in Fishkind et al., 2015, when relevant), the true model parameter $K = 3$ is used when implementing $\mathcal{L}^{P}$ (i.e. 3-means clustering is applied), and $\mathcal{K} = 4$ is used in Algorithm 1 when implementing $\mathcal{L}^{EP}$.

We first compare the effectiveness and runtime of $\mathcal{L}^{C}$ and $\mathcal{L}^{CS}$ in the small scale regime, which is the only scale on which $\mathcal{L}^{C}$ can be feasibly implemented. In implementing $\mathcal{L}^{CS}$ we used $nMCMC = 10000$, with $nMCMC/2 = 5000$ of these steps discarded as a burn-in. Results from the $nMC = 10000$ experiment realizations are summarized in Table 3 and Fig. 2.

Observe that $\mathcal{L}^{CS}$ obtains the optimal effectiveness of $\mathcal{L}^{C}$ while running orders of magnitude faster than $\mathcal{L}^{C}$; note that the running time of $\mathcal{L}^{CS}$ is relatively constant at each of the three small scale experiments while, empirically, the running of time $\mathcal{L}^{C}$ scales at rate about $2.6^{|A|}$; see Table 3. Indeed, $\mathcal{L}^{CS}$ can be efficiently implemented on graphs with hundreds of thousands of vertices while $\mathcal{L}^{C}$ cannot be practically implemented on graphs with more than a few tens of vertices. At this small scale, we did not include the spectral-based vertex nomination schemes $\mathcal{L}^{EP}$ and $\mathcal{L}^{P}$, because they are essentially ineffective at this small scale, since the eigenvectors contain almost no signal, as noted in Fishkind et al. (2015).

Next we move to the medium scale and large scale experiments, with stochastic block model parameters as given in Table 2. We did $nMC = 100$ experiment replicates for each of the vertex nomination schemes; $\mathcal{L}^{CS}$, $\mathcal{L}^{EP}$, $\mathcal{L}^{P}$, and we also included the likelihood maximization vertex nomination scheme $\mathcal{L}^{ML}$ introduced in Fishkind et al. (2015), since it was demonstrated in Fishkind et al. (2015) and Lyzinski et al. (2016) that $\mathcal{L}^{ML}$ obtains state-of-the-art effectiveness when implementable (i.e., for graphs of order at most a few thousand vertices). The canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$ was performed in two ways; once with $nMCMC = 100000$, and once with $nMCMC$ chosen to be such that the runtime of $\mathcal{L}^{CS}$ is equal to the runtime of $\mathcal{L}^{EP}$. The canonical vertex nomination scheme $\mathcal{L}^{C}$ was not performed in the medium scale and large scale, nor the likelihood maximization vertex nomination scheme $\mathcal{L}^{ML}$ at the large scale, because they are not practical to compute at these scales. The results of these simulations are summarized in Table 4 and in Fig. 3.

First, observe that in both the medium and the large scale $\mathcal{L}^{EP}$ was more effective than $\mathcal{L}^{P}$, significantly so in the medium scale regime, with a twofold runtime increase being the cost for this increase in effectiveness. In the adjacency spectral embedding of a stochastic block model, the within-class variance is, with high probability, of the order $\frac{\log n}{\sqrt{n}}$; see Lyzinski et al. (2014). Thus, as there are more vertices, the true clusters become more easily delineated, and the

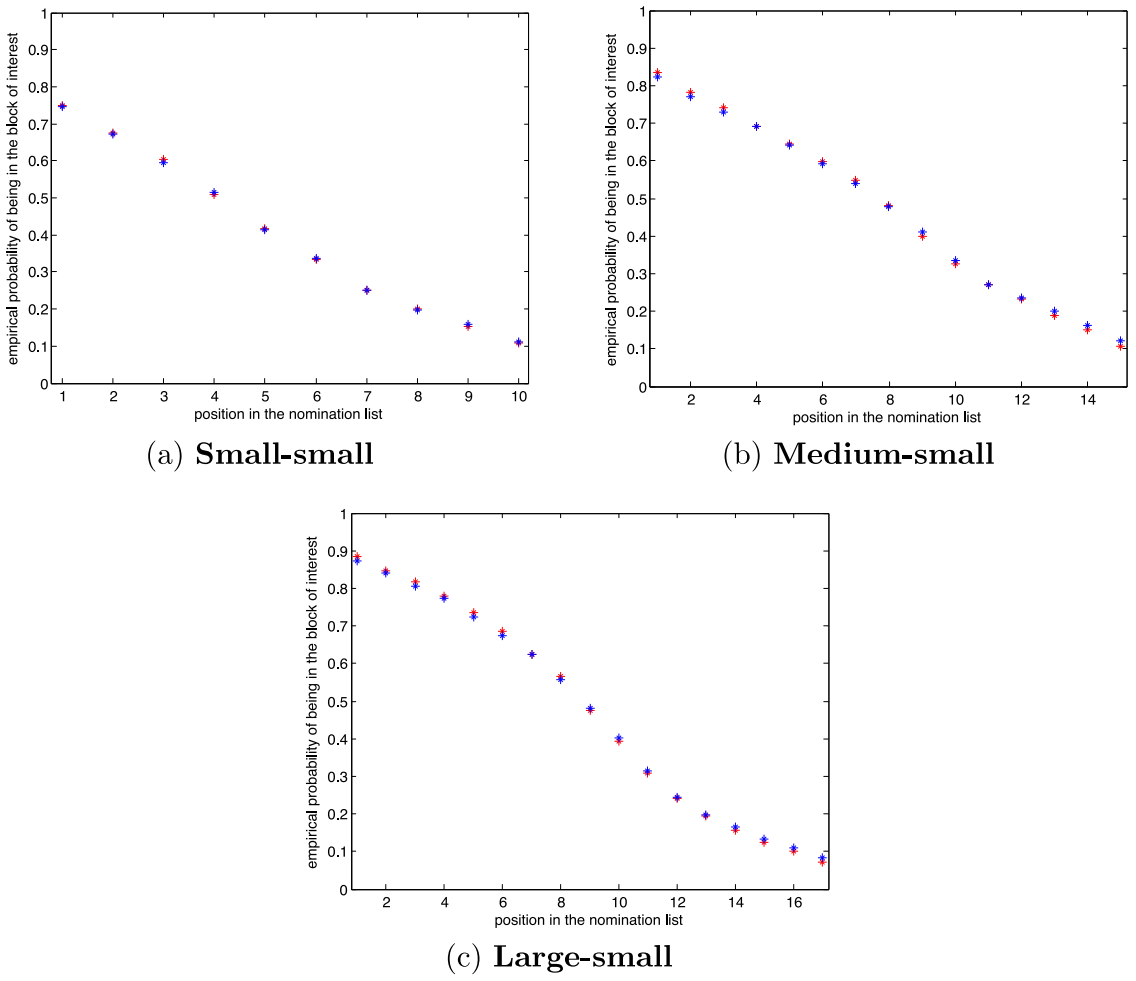


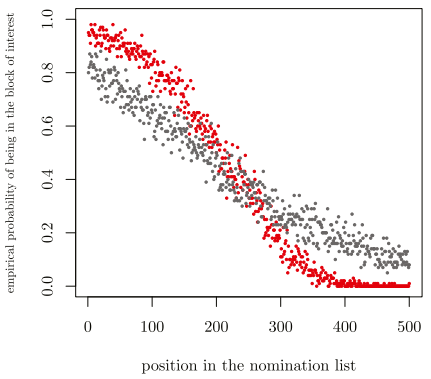
Fig. 2. Small scale simulations. Empirical probability of being in V_1 (y-axis) plotted against the respective position in the nomination list (x-axis) for $\mathcal{L}^C$ (red) and $\mathcal{L}^{CS}$ (blue). Here $nMC = 10000$, and for $\mathcal{L}^{CS}$ we use $nMCMC = 10000$; with $nMCMC/2 = 5000$ steps used for burn-in. (Note that some red asterisks in these figures are partially or nearly completely obscured by blue asterisks on top of them.).

Table 4
Medium and large scale experiments. Comparing $\mathcal{L}^P$, $\mathcal{L}^{EP}$, $\mathcal{L}^{CS}$ and $\mathcal{L}^{ML}$ by average runtime and empirical MAP.

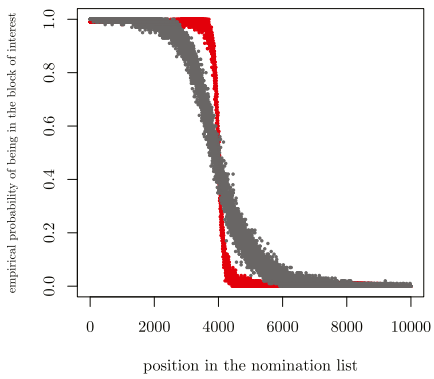
	$\mathcal{L}^P$	$\mathcal{L}^{EP}$	$\mathcal{L}^{CS}$; *nMCMC* set to match runtime of $\mathcal{L}^{EP}$	$\mathcal{L}^{CS}$; *nMCMC* set to 100000	$\mathcal{L}^{ML}$
Scale	Running time (in s)				
Medium	0.24	0.44	⟵ same	2.06	216.45
Large	19.42	19.57	⟵ same	112.51	*
Scale	MAP±2 s.e.				
Medium	$.74 \pm .02$	$.89 \pm .02$	$.80 \pm .01$	$.93 \pm .00$	$.95 \pm .00$
Large	$.99 \pm .02$	$.99 \pm .02$	$.66 \pm .00$	$.95 \pm .00$	*

adjacency spectral clustering step of $\mathcal{L}^{EP}$ and of $\mathcal{L}^P$ is dominated in running time by the embedding step, which is the same for $\mathcal{L}^{EP}$ and $\mathcal{L}^P$. However, in the medium scale regime, where the true clusters are less easily recovered in the embedding, the more sophisticated clustering procedure utilized in $\mathcal{L}^{EP}$ is significantly more effective than the k-means clustering used in $\mathcal{L}^P$—at the expense of an increase in runtime.

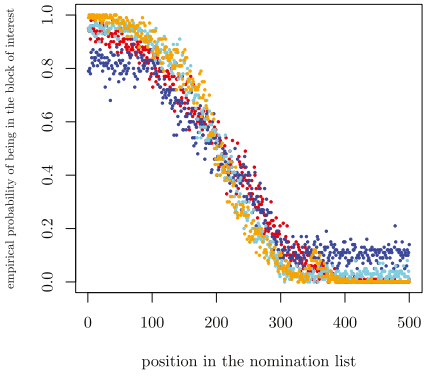
In the medium scale regime, while we see that $\mathcal{L}^{ML}$ is the most effective of the vertex nomination schemes that we compare, note that the runtime of $\mathcal{L}^{ML}$ was orders of magnitude greater then the other vertex nomination schemes. In fact, $\mathcal{L}^{ML}$ is not practical to implemented on graphs with more than a few thousand vertices (such as our large scale experiment), unlike $\mathcal{L}^{CS}$ and $\mathcal{L}^{EP}$. In both the medium and large scale examples, we see that $\mathcal{L}^{EP}$ is significantly more effective than $\mathcal{L}^{CS}$ when $\mathcal{L}^{CS}$ is restricted to have the same running time as $\mathcal{L}^{EP}$. However, $\mathcal{L}^{CS}$ will eventually be more



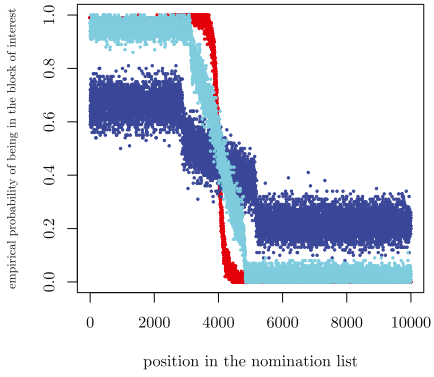
(a) **Medium scale:**
$\mathcal{L}^{EP}$ (red) versus $\mathcal{L}^{P}$ (grey).



(b) **Large scale:**
$\mathcal{L}^{EP}$ (red) versus $\mathcal{L}^{P}$ (grey).



(c) **Medium scale:**
$\mathcal{L}^{EP}$ (red); $\mathcal{L}^{CS}$ with $nMCMC$ chosen so that $\mathcal{L}^{EP}$ runtime is same (blue); $\mathcal{L}^{CS}$ with $nMCMC = 100000$ (cyan); $\mathcal{L}^{ML}$ (orange).



(d) **Large scale:**
$\mathcal{L}^{EP}$ (red); $\mathcal{L}^{CS}$ with $nMCMC$ chosen so that $\mathcal{L}^{EP}$ runtime is same (blue); $\mathcal{L}^{CS}$ with $nMCMC = 100000$ (cyan).

Fig. 3. Empirical probability of being in V_1 (y-axis) plotted against the respective position in the nomination list (x-axis) for the medium scale (left panels) and large scale (right panels) stochastic block model experiments.

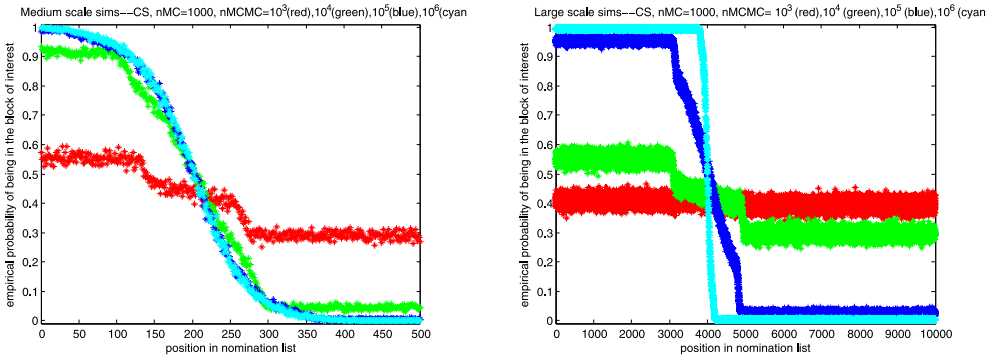
effective than $\mathcal{L}^{EP}$ (and all other vertex nomination schemes other than $\mathcal{L}^{C}$) given enough Markov chain Monte Carlo steps.

Indeed, to illustrate the effects of increasing the amount of sampling on $\mathcal{L}^{CS}$, we repeated the experiment in both the medium and large scales for $\mathcal{L}^{CS}$ with values $nMCMC = 10^3,\ 10^4,\ 10^5$, and 10^6. The results of $nMC = 1000$ realizations are shown in Fig. 4. In the medium scale, from $nMCMC = 10^4$ and up, the increased sampling still improved the effectiveness but seemed to stabilize towards a limit. In the large scale, continued steady improvement in effectiveness was seen for the increases in $nMCMC$, until $nMCMC = 10^6$ allowed for the near perfect success in the nomination task.

4.2. More simulation experiments

In this subsection, Section 4.2, we perform more simulation experiments to explore the tradeoff, for $\mathcal{L}^{CS}$ and $\mathcal{L}^{EP}$, between computational burden and effectiveness (i.e. precision). We also consider the effect of embedding dimension on the performance of $\mathcal{L}^{EP}$ since, in practice, the correct value of $d = \text{rank}\Lambda$ may not be known for use in the implementation of $\mathcal{L}^{EP}$.

In particular, in this subsection, the *embedding dimension* will refer to a positive integer $\eth$ that will replace d everywhere in the adjacency spectral embedding step of Section 3.3 (thus the vertices are embedded into $\mathbb{R}^{\eth}$ instead of $\mathbb{R}^{d}$)—and $\eth$ will also replace d onward in the definition of $\mathcal{L}^{EP}$ as given in Section 3.4. The results in Fishkind et al. (2013) imply that the effectiveness of $\mathcal{L}^{EP}$ should not degrade too much if $\eth > d$, but Fishkind et al. (2013) include an example (beginning of Section 8, see Fig. 1) where $\eth < d$ leads to a complete breakdown in spectral partitioning, with performance almost



(a) **Medium scale** (b) **Large scale**

Fig. 4. The effect on $\mathcal{L}^{CS}$ of increasing the value of *nMCMC*; plots are shown for $nMCMC = 10^3$ (red), $nMCMC = 10^4$ (green), $nMCMC = 10^5$ (blue), $nMCMC = 10^6$ (cyan).

Table 5
Trade-off of computational burden vs. precision between $\mathcal{L}^{EP}$ and $\mathcal{L}^{CS}$, and also comparison across different embedding dimensions. All times in this table are the average number of seconds, and all values of MAP are +/−.01. The runtimes in the bottom row−$\mathcal{L}^{CS}$ equiprecise time−have standard error ranging from .13 to .48, most are approximately .24. The runtimes in the second row−$\mathcal{L}^{EP}$ time−have standard error ranging from .01 to .09, most are approximately .04.

Embedding dimension $\eth$	2	3	4	5	8	9	10	11	12	15	20
$\mathcal{L}^{EP}$ MAP	.41	.53	.53	.51	.49	.50	.49	.49	.49	.48	.47
$\mathcal{L}^{EP}$ time	.50	.60	.84	1.01	1.71	2.02	2.37	2.72	3.09	4.39	6.31
$\mathcal{L}^{CS}$ equitime MAP	.13	.16	.25	.28	.33	.36	.39	.41	.44	.49	.56
$\mathcal{L}^{CS}$ equiprecise time	3.01	5.74	5.69	5.30	5.01	5.01	4.99	4.52	5.08	4.74	4.05

as bad as chance. In the setting we experiment with here, the effectiveness of $\mathcal{L}^{EP}$ will be seen as relatively robust to overestimation as well as underestimation of d.

Here we will use the following parameters: $K = 10$;

$$\Lambda = \begin{bmatrix} .30 & .27 & .24 & .21 & .21 & .21 & .21 & .21 & .21 & .21 \\ .27 & .30 & .27 & .24 & .21 & .21 & .21 & .21 & .21 & .21 \\ .24 & .27 & .30 & .27 & .24 & .21 & .21 & .21 & .21 & .21 \\ .21 & .24 & .27 & .30 & .27 & .24 & .21 & .21 & .21 & .21 \\ .21 & .21 & .24 & .27 & .30 & .27 & .24 & .21 & .21 & .21 \\ .21 & .21 & .21 & .24 & .27 & .30 & .27 & .24 & .21 & .21 \\ .21 & .21 & .21 & .21 & .24 & .27 & .30 & .27 & .24 & .21 \\ .21 & .21 & .21 & .21 & .21 & .24 & .27 & .30 & .27 & .24 \\ .21 & .21 & .21 & .21 & .21 & .21 & .24 & .27 & .30 & .27 \\ .21 & .21 & .21 & .21 & .21 & .21 & .21 & .24 & .27 & .30 \end{bmatrix}, \vec{n} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}, \vec{m} = \begin{bmatrix} 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \end{bmatrix}.$$

These parameters were chosen so that the blocks are stochastically similar to each other, there are many blocks, and the differences between the probabilities in Λ are mild relative to the number of vertices involved; all of these factors make the vertex nomination task quite challenging, since there is a limited amount of signal present.

For each value of embedding dimension $\eth = 2, 3, 4, 5, 8, 9, 10, 11, 12, 15, 20$ we obtained $nMC = 200$ independent realizations of the random graph with the above parameters, and we nominated for the block of interest V_1 using the extended spectral partitioning vertex nomination scheme $\mathcal{L}^{EP}$, and we recorded the mean runtime and the also the empirical mean average precision. We also used the canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$ on these realizations, but chose the number of Markov chain Monte Carlo steps *nMCMC* so that the runtime was the same ("equitimed") as the mean $\mathcal{L}^{EP}$ runtime; we recorded the empirical mean average precision from this "equitimed" $\mathcal{L}^{CS}$. We also used the canonical sampling vertex nomination scheme $\mathcal{L}^{CS}$ again on these realizations, but now we allowed the number of Markov Chain Monte Carlo steps *nMCMC* to be exactly as large as needed to achieve equal empirical mean average precision as was achieved by $\mathcal{L}^{EP}$; we recorded the mean runtime of this "equiprecise" $\mathcal{L}^{CS}$. (Because the value of *nMCMC* was not known a priori, we fixed the burn-in for $\mathcal{L}^{CS}$ in this subsection at $T = 5000$.) The results of these experiments are displayed in Table 5.

Note that when devoting the same computational resources to $\mathcal{L}^{CS}$ and $\mathcal{L}^{EP}$, we saw that here, for smaller values of $\eth$, $\mathcal{L}^{EP}$ achieved higher mean average precision than did $\mathcal{L}^{CS}$ and, for larger values of $\eth$, $\mathcal{L}^{CS}$ achieved higher mean average precision than did $\mathcal{L}^{EP}$. This is because $\mathcal{L}^{EP}$ took longer and longer to run in more dimensions, and the increased sampling time allowed $\mathcal{L}^{CS}$ to pull ahead in precision. Indeed, the mean average precision of $\mathcal{L}^{EP}$ is terminal, in contrast to $\mathcal{L}^{CS}$, for

Table 6
Comparison of MAP and runtimes for vertex nomination schemes on the connectome.

Nomination scheme	MAP	Avg. running time
$\mathcal{L}^{CS}$ "longer"	.86	93.02 s
$\mathcal{L}^{EP}$	.81	3.10 s
$\mathcal{L}^{P}$	.74	3.70 s
$\mathcal{L}^{CS}$ "shorter"	.60	2.81 s

which longer and longer sampling times will increase its mean average precision as long as patience allows—and, in the limit, to the highest attainable mean average precision.

Also note that the performance of $\mathcal{L}^{EP}$ here was relatively robust for incorrect embedding dimension ($\eth$ being greater or lesser then d). Although Fishkind et al. (2013) highlights by example the dangers of underestimating d, this example illustrates that such underestimation can be benign. In particular, $\eth = 3, 4$ led to somewhat better performance than the correct value $\eth = d = 10$. This can be explained by the decay in the eigenvalues of Λ; here the eigenvalues of Λ are 2.3465, 0.2197, 0.1745, 0.1112, 0.0648, 0.0300, 0.0235, 0.0178, 0.0064, 0.0056. After the first four greatest eigenvalues, the rest are small enough to cause Λ to produce behavior similar to that which a lower rank matrix would produce. Rigorous analysis of the optimal embedding dimension is beyond the scope of this present paper; see Yang et al. (2019) for principled methodology.

4.3. Real data example: A human connectome

In this subsection, Section 4.3, we consider a real-data example; a human connectome. This is a graph with vertices corresponding to locations in a human brain and edges which reflect functional adjacency. The block structure that we consider is not ostensibly reflective of an actual stochastic block model. Indeed, the vagarities of such real data gives us no reason to expect that there is precisely an underlying probabilistic block uniformity. Nonetheless, employing a stochastic block model as an approximation seems to be a plausibly useful approach. In fact, we will see that all of the important operational observations of this article do indeed occur here. Specifically, on this large graph, where L^{ML} and L^{C} schemes are not practical to implement, we will see that the nomination schemes introduced in this article scale very well, and we will see here that the extended spectral partitioning vertex nomination scheme is significantly more effective than the (original) spectral partitioning vertex nomination scheme, and the canonical sampling vertex nomination scheme is more effective than both—when enough computation is performed.

The human connectome (brain graph) that we use here comes from the very recent paper Kiar et al. (2017); the particular connectome that we employ is actually one level of a multiscale hierarchy provided there, and this hierarchy is sure to be a rich object of study in future work. Our graph was obtained as follows. Two diffusion MRI (dMRI) and two structural MRI (sMRI) scans were done on an individual, collected over two sessions (Zuo et al., 2014). Graphs were estimated using the NDMG (Zuo et al., 2014) pipeline. The dMRI scans were pre-processed for eddy currents using FSL's `eddy_correct` (Andersson et al., 2003). FSL's "standard" linear registration pipeline was used to register the sMRI and dMRI images to the MNI152 atlas (Smith et al., 2004; Woolrich et al., 2009; Jenkinson et al., 2012; Mazziotta et al., 2001). A tensor model was fit using DiPy (Garyfallidis et al., 2014) to obtain an estimated tensor at each voxel. A deterministic tractography algorithm was applied using DiPy's EuDX (Garyfallidis et al., 2014, 2012) to obtain a fiber streamline from each voxel. Graphs were formed by contracting fiber streamlines into sub-regions depending on spatial (Mhembere et al., 2013) proximity or neuro-anatomical (Tzourio-Mazoyer et al., 2002; Desikan et al., 2006; Makris et al., 2006; Lancaster, 1997; Oishi et al., 2010; Glasser et al., 2016; Wang et al., 2014; Sripada et al., 2014; Kessler et al., 2014) similarity.

We consider a three block SBM model for this data; V_1 are the regions corresponding to the right hemisphere, V_2 are the regions corresponding to the left hemisphere, and V_3 are regions that are not characterized. In particular, $n_1 = 2807$, $n_2 = 2780$, and $n_3 = 271$. The number of seeds we considered were $m_1 = 500$, $m_2 = 500$, $m_3 = 50$, respectively; in each of $nMC = 500$ experiment replicates, we independently discrete-uniformly selected the seeds from the blocks, and constructed a nomination list for the remaining 4808 ambiguous vertices using each of vertex nomination schemes $\mathcal{L}^{P}$, $\mathcal{L}^{EP}$, "shorter" $\mathcal{L}^{CS}$, and "longer" $\mathcal{L}^{CS}$. "Longer" $\mathcal{L}^{CS}$ used $nMCMC = 100000$ and "shorter" $\mathcal{L}^{CS}$ used $nMCMC = 3000$, the latter value chosen so that $\mathcal{L}^{CS}$ runtime was approximately the same as the runtime of $\mathcal{L}^{EP}$. Both $\mathcal{L}^{P}$ and $\mathcal{L}^{EP}$ used embedding dimension $d = 6$ (since this was the first elbow in the scree plot as determined through the algorithm of Zhu and Ghodsi, 2006); $\mathcal{L}^{P}$ used 1000 k-means restarts, and $\mathcal{L}^{EP}$ considered the 'EEV', 'EEE', and 'EII' covariance structures in Table 1, and $\mathcal{K} = 3$ number of clusters. For each of "shorter" $\mathcal{L}^{CS}$ and "longer" $\mathcal{L}^{CS}$, the value of Λ was estimated from population densities, and half of $nMCMC$ steps were burn-in.

The results of these experiments are summarized in Table 6 and Fig. 5. In particular, note that $\mathcal{L}^{EP}$ was substantially more effective than $\mathcal{L}^{P}$, although their runtimes were about the same. Also note that when $\mathcal{L}^{CS}$ was limited in runtime to the order of runtime for $\mathcal{L}^{EP}$, it was not competitive in terms of effectiveness but, with increased runtime, $\mathcal{L}^{CS}$ did eventually overtake all of the other vertex nomination schemes in terms of effectiveness. On a graph of this order, having approximately 5000 ambiguous vertices, the likelihood maximization vertex nomination scheme $\mathcal{L}^{ML}$ and the canonical vertex nomination scheme $\mathcal{L}^{C}$ were not tractable. Indeed, these experiments highlight the scalability and effectiveness of the vertex nomination schemes $\mathcal{L}^{CS}$ and $\mathcal{L}^{EP}$ introduced in this paper.

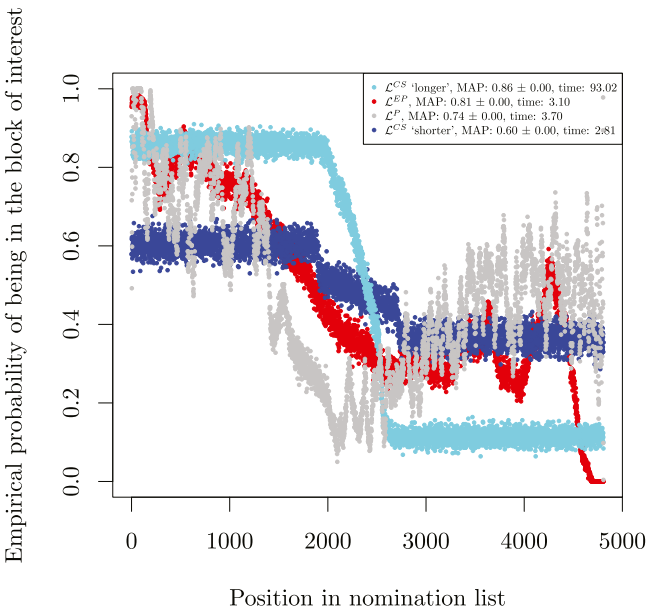


Fig. 5. For the connectome real-data experiments, comparing the effectiveness of $\mathcal{L}^{P}$ (gray), $\mathcal{L}^{EP}$ (red), "shorter" $\mathcal{L}^{CS}$ (blue), and "longer" $\mathcal{L}^{CS}$ (cyan).

5. Summary and future directions

In summary, for a vertex nomination instance, the optimally precise vertex nomination scheme – the canonical nomination scheme $\mathcal{L}^{C}$ – is only practical for the smallest, toy problems. For larger instances, the likelihood maximization nomination scheme $\mathcal{L}^{ML}$ should be used, until the size of the problem is too big for this to be practical, which may be on the order of a thousand or so vertices. For larger instances, the extended spectral partitioning $\mathcal{L}^{EP}$ and the canonical sampling $\mathcal{L}^{CS}$ vertex nomination schemes (introduced in this paper) should be used; the former can be the better choice when computational resources are more limited and less is known about the model parameters, and the latter can be the better choice when there is more knowledge of the model parameters and there are greater computational resources.

Concurrent work in vertex nomination has tackled the nomination problem in a slightly modified setting, considering a pair of networks and using vertices of interest in one network to nominate potential vertices of interest in the second network (Patsolic et al., 2017; Lyzinski et al., 2019). In this paired graph setting, the concept of nomination consistency is established for general network models (and for a general notion of "vertices of interest") in Lyzinski et al. (2019) and Agterberg et al. (2019), and the surprising fact that universally consistent vertex nomination schemes do not exist is established in Lyzinski et al. (2019). In the present, single network setting, this points to a direction for future research: Generalizing the concept of vertices of interest beyond community membership, and establishing the statistical framework for vertex nomination consistency in the setting where more general vertex covariates delineate "interesting" versus "non-interesting" vertices.

Acknowledgments

The authors are grateful to the referees and editors for very useful feedback that greatly enhanced this paper. Support in part provided by the Johns Hopkins University Human Language Technology CoE, the DARPA SIMPLEX program through contract N66001-15-C-4041, the DARPA D3M program through contract FA8750-17-2-0112, and the Acheson J. Duncan Fund for the Advancement of Research in Statistics at Johns Hopkins University. The work of authors JY, LC, HP was undertaken while graduate students at Johns Hopkins University.

References

Agterberg, J., Park, Y., Larson, J., White, C., Priebe, C.E., Lyzinski, V., 2019. Vertex nomination, consistent estimation, and adversarial modification. arXiv preprint arXiv:1905.01776.

Airoldi, E.M., Blei, D.M., Fienberg, S.E., 2008. Mixed membership stochastic blockmodels. J. Mach. Learn. Res. 9, 1981–2014.

Aldous, D., Fill, J.A., 2002. Reversible Markov Chains and Random Walks on Graphs, Berkeley.

Andersson, J.L.R., Skare, S., Ashburner, J., 2003. How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging. NeuroImage 20, 870–888.

Athreya, A., Lyzinski, V., Marchette, D.J., Priebe, C.E., Sussman, D.L., Tang, M., 2015. A limit theorem for scaled eigenvectors of random dot product graphs. Sankhya.

Bickel, P., Choi, D., Chang, X., Zhang, H., 2013. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. Ann. Statist. 41:4, 1922–1943.
Chatterjee, S., 2014. Matrix estimation by universal singular value thresholding. Ann. Statist. 43:1, 177–214.
Coppersmith, G., 2014. Vertex nomination. Wiley Interdiscip. Rev. Comput. Stat. 6:2, 144–153.
Coppersmith, G.A., Priebe, C.E., 2012. Vertex nomination via content and context. arXiv preprint arXiv:1201.4118.
Desikan, R.S., Segonne, F., Fischl, B., Quinn, B.T., Dickerson, B.C., Blacker, D., Buckner, R.L., Dale, A.M., Maquire, R.P., Hyman, B.T., Albert, M.S., Killany, R.J., 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. NeuroImage.
Diaconis, P., Saloff-Coste, L., 1998. What do we know about the metropolis algorithm? J. Comput. System Sci. 57, 20–36.
Erdos, P., Renyi, A., 1963. Asymmetric graphs. Acta Math. Acad. Sci. Hung. 14, 295–315.
Feller, W., 2008. An Introduction to Probability Theory and Its Applications, Vol. 2. John Wiley & Sons.
Fishkind, D.E., Lyzinski, V., Pao, H., Chen, L., Priebe, C.E., 2015. Vertex nomination schemes for membership prediction. Ann. Appl. Stat. 9:3, 1510–1532.
Fishkind, D.E., Sussman, D.L., Tang, M., Vogelstein, J.T., Priebe, C.E., 2013. Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown. SIAM J. Matrix Anal. Appl. 34, 23–39.
Fraley, C., Raftery, A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. J. Amer. Statist. Assoc. 97, 611–631.
Fraley, C., Raftery, A.E., 2006. MCLUST version 3: an R package for normal mixture modeling and model-based clustering, DTIC Document.
Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., Van Der Walt, S., Descoteaux, M., Nimmo-Smith, I., 2014. Dipy, a library for the analysis of diffusion MRI data. Front. Neuroinformatics 8, 8.
Garyfallidis, E., Brett, M., Correia, M.M., Williams, G.B., Nimmo-Smith, I., 2012. Quickbundles, a method for tractography simplification. Front. Neurosci. 6, 175.
Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 2014. Bayesian Data Analysis, Vol. 2. Taylor & Francis.
Gilks, W.R., Richardson, S., Spiegelhalter, D., 1995. Markov Chain Monte Carlo in Practice. CRC press.
Glasser, M.F., Coalson, T.S., Robinson, E.C., Hacker, C.D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C.F., Jenkinson, M., Smith, S.M., Van Essen, D.C., 2016. A multi-modal parcellation of human cerebral cortex. Nature 7615, 171–178.
Hardy, G.H., Littlewood, J.E., Polya, G., 1952. Inequalities, second ed. Cambridge University Press.
Holland, P.W., Laskey, K., Leinhardt, S., 1983. Stochastic blockmodels: First steps. Social Networks 5:2, 109–137.
Jenkinson, M., Beckmann, C.F., Behrens, T.E., Woolrich, M.W., Smith, S.M., 2012. FSL. NeuroImage 2, 782–790.
Johndrow, J.E., Smith, A., 2018. Fast mixing of metropolis-hastings with unimodal targets. Electron. Commun. Probab. 23, 1–9.
Karrer, B., Newman, M.E.J., 2011. Stochastic blockmodels and community structure in networks. Phys. Rev. E 83.
Kaufman, L., Rousseeuw, P.J., 2009. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons.
Kessler, D., Angstadt, M., Welsh, R.C., Sripada, C., 2014. Modality-spanning deficits in attention-deficit/hyperactivity disorder in functional networks, gray matter, and white matter. J. Neurosci. 34, 16555–16566.
Kiar, G., Bridgeford, E.W., Roncal, W.G., Consortium for Reliability and Reproducibility (CoRR), Chandrashekhar, V., Mhembere, D., Ryman, S., Zuo, X., Margulies, D.S., Craddock, R.C., Priebe, C.E., Jung, R., Calhoun, V.D., Caffo, B., Burns, R., Milham, M.P., Vogelstein, J.T., 2017. A principled high-throughput estimation and mega-analysis pipeline for reproducible connectomics. Preprint available at https://www.biorxiv.org/content/early/2017/09/14/188706.
Lancaster, J.L., 1997. The Talairach Daemon, a database server for Talairach atlas labels. NeuroImage.
Lyzinski, V., Levin, K., Fishkind, D.E., Priebe, C.E., 2016. On the consistency of the likelihood maximization vertex nomination scheme: Bridging the gap between maximum likelihood estimation and graph matching. J. Mach. Learn. Res. 17, 1–34.
Lyzinski, V., Levin, K., Priebe, C.E., 2019. On consistent vertex nomination schemes. J. Mach. Learn. Res. 20 (69), 1–39.
Lyzinski, V., Sussman, D.L., Tang, M., Athreya, A., Priebe, C.E., 2014. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. Electron. J. Stat. 8, 2905–2922.
MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, (14), pp. 281–297.
Makris, N., Goldstein, J.M., Kennedy, D., Hodge, S.M., Caviness, V.S., Faraone, S.V., Tsuang, M.T., Seidman, L.J., 2006. Decreased volume of left and total anterior insular lobule in schizophrenia. Schizophr. Res. 83 (2), 155–171.
Marchette, D., Priebe, C.E., Coppersmith, G., 2011. Vertex nomination via attributed random dot product graphs. In: Proceedings of the 57th ISI World Statistics Congress, vol. 6.
Mazziotta, J., Toga, A., Evans, A., Fox, P., Lancaster, J., Zilles, K., Woods, R., Paus, T., Simpson, G., Pike, B., Holmes, C., Collins, L., Thompson, P., MacDonald, D., Iacoboni, M., Schormann, T., Amunts, K., Palomero-Gallagher, N., Geyer, S., Parsons, L., Narr, K., Kabani, N., Le Goualher, G., Feidler, J., Smith, K., Boomsma, D., Hulshoff Pol, H., Cannon, T., Kawashima, R., Mazoyer, B., 2001. A four-dimensional probabilistic atlas of the human brain. J. Am. Med. Inform. Assoc. 8 (5), 401–430.
McLachlan, G., Peel, D., 2004. Finite Mixture Models. John Wiley & Sons.
Mhembere, D., Roncal, W.G., Sussman, D.L., Priebe, C.E., Jung, R., Ryman, S., Vogelstein, R.J., Vogelstein, J.T., Burns, R., 2013. Computing scalable multivariate glocal invariants of large (brain-) graphs. In: IEEE Global Conference on Signal and Information Processing, GlobalSIP, pp. 297–300.
Newman, M.E.J., 2006. Modularity and community structure in networks. Proc. Natl. Acad. Sci. 103:23, 8577–8582.
Oishi, K., Faria, A.V., van Zijl, P.C.M., Mori, S., 2010. MRI Atlas of Human White Matter. Academic Press.
Olhede, S.C., Wolfe, P.J., 2014. Network histograms and universality of block model approximation. Proc. Natl. Acad. Sci. 111, 14722–14727.
Patsolic, H.G., Park, Y., Lyzinski, V., Priebe, C.E., 2017. Vertex nomination via seeded graph matching. ArXiv preprint available at arXiv:1705.00674.
Polya, G., 1937. Kombinatorische anzahlbestimmungen für gruppen, graphen und chemische verbindungen. Acta Math. 68, 145–254.
Qin, T., Rohe, K., 2013. Regularized spectral clustering under the degree-corrected stochastic blockmodel. Adv. Neural Inf. Process. Syst..
Rohe, K., Chatterjee, S., Yu, B., 2011. Spectral clustering and the high-dimensional stochastic blockmodel. Ann. Statist. 39, 1878–1915.
Smith, S.M., Jenkinson, M., Woolrich, M.W., Beckmann, C.F., Behrens, T.E., Johansen-Berg, H., Bannister, P.R., De Luca, M., Drobnjak, I., Flitney, D.E., Niazy, R.K., Saunders, J., Vickers, J., Zhang, Y., De Stefano, N., Brady, J.M., Matthews, P.M., 2004. Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage 23, S208–219.
Sripada, C.S., Kessler, D., Angstadt, M., 2014. Lag in maturation of the brain's intrinsic functional architecture in attention-deficit/hyperactivity disorder. Proc. Natl. Acad. Sci. 111:39, 14259–14264.
Sun, M., Tang, M., Priebe, C.E., 2012. A comparison of graph embedding methods for vertex nomination. In: 2012 International Conference on Machine Learning and Applications, pp. 398–403.
Sussman, D.L., Tang, M., Fishkind, D.E., Priebe, C.E., 2012. A consistent adjacency spectral embedding for stochastic blockmodel graphs. J. Amer. Statist. Assoc. 107, 1119–1128.
Sussman, D.L., Tang, M., Priebe, C.E., 2014. Consistent latent position estimation and vertex classification for random dot product graphs. IEEE Trans. Pattern Anal. Mach. Intell. 36, 48–57.
Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., Joliot, M., 2002. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. Neuroimage 15:1, 273–289.
Von Luxburg, U., 2007. A tutorial on spectral clustering. Stat. Comput. 17:4, 395–416.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., 2001. Constrained k-means clustering with background knowledge, In: International Conference on Machine Learning, pp. 577–584.
Wang, Y.X.R., Bickel, P.J., 2017. Likelihood-based model selection for stochastic block models. Ann. Statist. 45:2, 500–528.
Wang, L., Mruczek, R.E.B., Michael, J., Kastner, S., 2014. Probabilistic maps of visual topography in human cortex. Cerebral Cortex 1–21.
Wang, Y.J., Wong, G.Y., 1987. Stochastic blockmodels for directed graphs. J. Amer. Statist. Assoc. 82, 8–19.
Wolfe, P., Olhede, S.C., 2013. Nonparametric graphon estimation. ArXiv preprint available at http://arxiv.org/abs/1309.5936.
Woolrich, M.W., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., Beckmann, C., Jenkinson, M., Smith, S.M., 2009. Bayesian analysis of neuroimaging data in FSL. NeuroImage 45, S173–186.
Yang, C., Priebe, C.E., Park, Y., Marchette, D.J., 2019. Simultaneous dimensionality and complexity model selection for spectral graph clustering. ArXiv preprint available at arXiv:1904.02926.
Yoder, J., 2016. On model-based semi-supervised clustering (Ph.D. dissertation). Johns Hopkins University.
Yoder, J., Priebe, C.E., 2014. A model-based semi-supervised clustering methodology. arXiv preprint arXiv:1412.4841.
Yoder, J., Priebe, C.E., 2017. Semi-supervised k-means++. J. Stat. Comput. Simul. 87, 2597–2608.
Zhu, M., Ghodsi, A., 2006. Automatic dimensionality selection from the scree plot via the use of profile likelihood. Comput. Statist. Data Anal. 51:2, 918–930.
Zuo, X.N., Anderson, J.S., Bellec, P., Birn, R.M., Biswal, B.B., Blautzik, J., Breitner, J.C., Buckner, R.L., Calhoun, V.D., Castellanos, F.X., Chen, A., Chen, B., Chen, J., Chen, X., Colcombe, S.J., Courtney, W., Craddock, R.C., Di Martino, A., Dong, H.M., Fu, X., Gong, Q., Gorgolewski, K.J., Han, Y., He, Y., He, Y., Ho, E., Holmes, A., Hou, X.H., Huckins, J., Jiang, T., Jiang, Y., Kelley, W., Kelly, C., King, M., LaConte, S.M., Lainhart, J.E., Lei, X., Li, H.J., Li, K., Li, K., Lin, Q., Liu, D., Liu, J., Liu, X., Liu, Y., Lu, G., Lu, J., Luna, B., Luo, J., Lurie, D., Mao, Y., Margulies, D.S., Mayer, A.R., Meindl, T., Meyer, M.E., Nan, W., Nielsen, J.A., O'Connor, D., Paulsen, D., Prabhakaran, V., Qi, Z., Qiu, J., Shao, C., Shehzad, Z., Tang, W., Villringer, A., Wang, H., Wang, K., Wei, D., Wei, G.X., Weng, X.C., Wu, X., Xu, T., Yang, N., Yang, Z., Zang, Y.F., Zhang, L., Zhang, Q., Zhang, Z., Zhang, Z., Zhao, K., Zhen, Z., Zhou, Y., Zhu, X.T., Milham, M.P., 2014. An open science resource for establishing reliability and reproducibility in functional connectomics. Sci. Data 1, 140049.