Connecting the Popularity Adjusted Block Model to the Generalized Random Dot Product Graph for Clustering and Parameter Estimation

John Koo, Minh Tang, Michael Trosset

Abstract

In this paper, we connect two random graph models, the Popularity Adjusted Block Model (PABM) and the Generalized Random Dot Product Graph (GRDPG) and use properties established in this connection to aid in community detection and parameter estimation. In particular, we note that the PABM can be represented as latent positions such that points within the same community lie on a subspace, and the subspaces that represent each community are orthogonal to one another. Using this property as well as the asymptotic properties of Adjacency Spectral Embedding (ASE) of the GRDPG, we are able to establish theoretical asymptotic results of our community detection and parameter estimation methods for the PABM.

1 Introduction

Statistical analysis on graphs or networks often involves the partitioning of a graph into disconnected subgraphs or clusters. This is often motivated by the assumption that there exist underlying and unobserved communities to which each vertex of the graph belongs, and edges between pairs of vertices are determined by drawing from a probability distribution based on the community relationships between each pair. The goal of the analysis then is population community detection, or the recovery of the true underlying community labels for each vertex, up to permutation (with some additional parameter estimation being of possible interest), assuming some underlying probability model. One such model is the Stochastic Block Model (SBM), first proposed by Lorrain and White [10], which assumes that the edge probability from one vertex to another follows a Beronulli distribution with fixed probabilities for each pair of community labels. Other random graph models have been proposed and studied, such as the Degree-Corrected Block Model (DCBM), introduced by Karrer and Newman [8], which is a generalization of the SBM. The Popularity Adjusted Block Model (PABM) was then introduced by Sengupta and Chen [15] as a generalization of the DCBM to address the heterogeneity of edge probabilities within and between communities while still maintaining distinct community structure.

The underlying similarity among the SBM, PABM, and other such models is that they involve a symmetric edge probability matrix $P \in [0,1]^{n \times n}$ where n is the number of vertices in

the graph. An undirected and unweighted graph is then drawn from this edge probability matrix such that the existence of an edge between each pair of vertices i and j is given by Bernoulli(P_{ij}). For example, for the SBM with two communities for which the within-community edge probability is ξ and the between-community edge probability is η, the entries of P consist of ξ and η.

The Random Dot Product Graph (RDPG) model proposed by Young and Scheinerman [20] is another graph model with Bernoulli edge probabilities. Under this model, each vertex of the graph can be represented by a point in some latent space such that the edge probability between any pair of vertices is given by their corresponding dot product in the latent space, i.e., given a latent positions $x_1, ..., x_n \in \mathbb{R}^d$, the edge probability matrix is $P = XX^\top$ where $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$. The SBM is equivalent to a special case of the RDPG model in which all vertices of a given community share the same position in the latent space [11]. It has also been shown that similar random graph models, including the DCBM, can be represented in this way [14] [11]. An analogous property exists for the PABM but not for the RDPG model but under the *Generalized* Random Dot Product Graph (GRDPG) model. This relationship will be explored in this paper and exploited to construct algorithms for community detection and parameter estimation for the PABM.

In this paper, we will only consider undirected graphs, that is the edge weight from vertex i to vertex j is equal to the edge weight in the opposite direction, from vertex j to vertex i. Furthermore, we will only consider unweighted graphs with binary $(0, 1)$ edge weights We will also assume that graphs are hollow, i.e., there are no edges from a vertex to itself. All such graphs can be represented by a symmetric adjacency matrix $A \in \{0, 1\}^{n\times n}$ for which $A_{ij} = 1$ if there exists an edge between vertices i and j and 0 otherwise, and A is an element-wise independent Bernoulli draw from a symmetric edge probability matrix $P \in [0, 1]^{n\times n}$.

2 Connecting the Popularity Adjusted Block Model to the Generalized Random Dot Product Graph

2.1 The popularity adjusted block model (PABM) and the generalized random dot product graph

Definition 1 (Popularity Adjusted Block Model) [15]. Let $P \in [0, 1]^{n\times n}$ be a symmetric edge probability matrix for a set of n vertices, V. Each vertex has a community label $1, ..., K$, and the rows and columns of P are arranged by community label such that $n_k \times n_l$ block $P^{(kl)}$ describes the edge probabilities between vertices in communities k and l ($P^{(lk)} = (P^{(kl)})^\top$). Let graph $G = (V, E)$ be an undirected, unweighted graph such that its corresponding adjacency matrix $A \in \{0, 1\}^{n\times n}$ is a realization of Bernoulli(P), i.e., $A_{ij} \overset{\text{indep}}{\sim}$ Bernoulli(P_{ij}) for $i > j$ ($A_{ij} = A_{ji}$ and $A_{ii} = 0$).

If each block $P^{(kl)}$ can be written as the outer product of two vectors:

$$P^{(kl)} = \lambda^{(kl)}(\lambda^{(lk)})^\top \tag{1}$$

for a set of K^2 fixed vectors $\{\lambda^{(st)}\}_{s,t=1}^{K}$ where each $\lambda^{(st)}$ is a column vector of dimension n_s, then graph G and its corresponding adjacency matrix A is a realization of a popularity adjusted block model with parameters $\{\lambda^{(st)}\}_{s,t=1}^{K}$.

We will use the notation $A \sim PABM(\{\lambda^{(kl)}\}_K)$ to denote a random adjacency matrix A drawn from a PABM with parameters $\lambda^{(kl)}$ consisting of K underlying communities.

Definition 2 (Generalized Random Dot Product Graph) [13]. Let $P \in [0,1]^{n\times n}$ be a symmetric edge probability matrix for a set of n vertices, V. If $\exists X \in \mathbb{R}^{n\times d}$ such that

$$P = XI_{pq}X^\top \tag{2}$$

for some $d, p, q \in \mathbb{N}$ and $p+q=d$, then graph $G=(V,E)$ with adjacency matrix A such that $A_{ij} \overset{\text{indep}}{\sim} \text{Bernoulli}(P_{ij})$ for $i>j$ ($A_{ij}=A_{ji}$ and $A_{ii}=0$) is a draw from the generalized random dot product graph model with latent positions X and signature (p,q). More precisely, if vertices i and j have latent positions x_i and x_j respectively, then the edge probability between the two is $P_{ij}=x_i^\top I_{pq}x_j$, and X contains the latent positions as rows $x_i^\top$.

We will use the notation $A \sim GRDPG_{p,q}(X)$ to denote a random adjacency matrix A drawn from latent positions X and signature (p,q).

Definition 3. The indefinite orthogonal group with signature (p,q) is the set $\{Q \in \mathbb{R}^{d\times d} : QI_{pq}Q^\top = I_{pq}\}$, denoted as $\mathbb{O}(p,q)$ [13].

Remark. Like the RDPG, the latent positions of a GRDPG are not unique [13]. More specifically, if $P_{ij}=x_i^\top I_{pq}x_j$, then we also have for any $Q \in \mathbb{O}(p,q)$, $(Qx_i)^\top I_{pq}(Qx_j) = x_i^\top(Q^\top I_{pq}Q)x_j = x_i^\top I_{pq}x_j = P_{ij}$. Unlike in the RDPG case, transforming the latent positions via multiplication by $Q \in \mathbb{O}(p,q)$ does not necessarily maintain interpoint angles or distances.

2.2 Connecting the PABM to the GRDPG

Theorem 1 (Connecting the PABM to the GRDPG for $K=2$). Let

$$X = \begin{bmatrix} \lambda^{(11)} & \lambda^{(12)} & 0 & 0 \\ 0 & 0 & \lambda^{(21)} & \lambda^{(22)} \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

as in Definition 1. Then $A \sim GRDPG_{3,1}(XU)$ and $A \sim PABM(\{(\lambda^{(kl)}\}_2)$ are equivalent.

Theorem 2 (Generalization to $K>2$). There exists a block diagonal matrix $X \in \mathbb{R}^{n\times K^2}$ defined by PABM parameters $\{\lambda^{(kl)}\}_K$ and $U \in \mathbb{R}^{K^2\times K^2}$ that is fixed for each K such that $A \sim GRDPG_{K(K+1)/2,K(K-1)/2}(XU)$ and $A \sim PABM(\{(\lambda^{(kl)}\})_K)$ are equivalent.

Example ($K = 3$). Using the same notation as before:

$$X = \begin{bmatrix} \lambda^{(11)} & \lambda^{(12)} & \lambda^{(13)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda^{(21)} & \lambda^{(22)} & \lambda^{(23)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda^{(31)} & \lambda^{(32)} & \lambda^{(33)} \end{bmatrix}$$

$$Y = \begin{bmatrix} \lambda^{(11)} & 0 & 0 & \lambda^{(12)} & 0 & 0 & \lambda^{(13)} & 0 & 0 \\ 0 & \lambda^{(21)} & 0 & 0 & \lambda^{(22)} & 0 & 0 & \lambda^{(23)} & 0 \\ 0 & 0 & \lambda^{(31)} & 0 & 0 & \lambda^{(32)} & 0 & 0 & \lambda^{(33)} \end{bmatrix}$$

Then $P = XY^\top$ and $Y = X\Pi$ where Π is a permutation matrix consisting of 3 fixed points and 3 cycles of order 2:

$$\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Positions 1, 5, 9 are fixed.
- The cycles of order 2 are $(2, 4)$, $(3, 7)$, and $(6, 8)$.

Therefore, we can decompose $\Pi = UI_{6,3}U^\top$ where the first three columns of U consist of e_1, e_5, and e_9 corresponding to the fixed positions 1, 5, and 9, the next three columns consist of eigenvectors $(e_k + e_l)/\sqrt{2}$, and the last three columns consist of eigenvectors $(e_k - e_l)/\sqrt{2}$, where pairs (k, l) correspond to the cycles of order 2 described above.

The latent positions are the rows of

$$XU = \begin{bmatrix} \lambda^{(11)} & 0 & 0 & \lambda^{(12)}/\sqrt{2} & \lambda^{(13)}/\sqrt{2} & 0 & \lambda^{(12)}/\sqrt{2} & \lambda^{(13)}/\sqrt{2} & 0 \\ 0 & \lambda^{(22)} & 0 & \lambda^{(21)}/\sqrt{2} & 0 & \lambda^{(23)}/\sqrt{2} & -\lambda^{(21)}/\sqrt{2} & 0 & \lambda^{(23)}/\sqrt{2} \\ 0 & 0 & \lambda^{(33)} & 0 & \lambda^{(31)}/\sqrt{2} & \lambda^{(32)}/\sqrt{2} & 0 & -\lambda^{(31)}/\sqrt{2} & -\lambda^{(32)}/\sqrt{2} \end{bmatrix}$$

3 Methods

Two inference objectives arise from the PABM:

1. Community membership identification (up to permutation).
2. Parameter estimation (estimating $\lambda^{(kl)}$'s).

In our methods, we assume that K, the number of communities, is known beforehand and does not require estimation.

3.1 Related work

Sengupta and Chen [15], who first proposed the PABM, used a relaxed version of Modularity Maximization (MM) based on the Extreme Points (EP) algorithm [9] for community detection and parameter estimation. They were able to show that as the sample size increases, the proportion of misclassified community labels (up to permutation) goes to 0.

Noroozi, Rimal, and Pensky [12] used Sparse Subspace Clustering (SSC) for community detection in the PABM. SSC is performed by solving an optimization problem for each observed point. Given $X \in \mathbb{R}^{n \times d}$ with vectors $x_i^\top \in \mathbb{R}^d$ as rows of X, the optimization problem $\min_{c_i} ||c_i||_1$ subject to $x_i = Xc_i$ and $\beta_i = 0$ is solved for each $i = 1, ..., n$. The solutions are collected collected into matrix $C = \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}^\top$ to construct an affinity matrix $B = |C| + |C^\top|$. If each x_i lie perfectly on one of K subspaces, B describes an undirected graph consisting of K disjoint subgraphs, i.e., $B_{ij} = 0$ if x_i, x_j are in different subspaces. If X instead represents points near K subspaces with some noise, a final graph partitioning step is performed (e.g., edge thresholding or spectral clustering).

In practice, SSC is often performed by solving the LASSO problems

$$\min_{c_i} \frac{1}{2}||x_i - X_{-i}c_i||_2^2 + \lambda||c_i||_1 \tag{3}$$

for some sparsity parameter $\lambda > 0$. The c_i vectors are then collected into C and B as before.

Theorem 2 suggests that SSC is appropriate for community detection for the PABM. More precisely, Theorem 2 says that each community consists of a K-dimensional subspace, and together the subspaces lie in $\mathbb{R}^{K^2}$. The natural approach then is to perform SSC on the ASE of P or A. Noroozi et al. [12] instead applied SSC to P and A themselves.

Much of our work is built on top of work from Rubin-Delanchy et al. [13], in particular, the properties of the ASE of A in relation to the ASE of P built on $\{\lambda^{(kl)}\}_K$.

3.2 Community detection

We previously stated one possible set of latent positions that result in the edge probability matrix of a PABM, $P = (XU)I_{pq}(XU)^\top$. If we have (or can estimate) XU directly, then both the community detection and parameter identification problem are trivial since U is orthonormal and fixed for each value of K. However, direct identification or estimation of XU is not possible [13].

If we decompose $P = ZI_{pq}Z^\top$, then $\exists Q \in \mathbb{O}(p, q)$ such that $XU = ZQ$. Even if we start with the exact edge probability matrix, we cannot recover the "original" latent positions XU. Note that unlike in the case of the RDPG, Q is not an orthogonal matrix. If z_i's are the rows of XU, then $||z_i - z_j||^2 \neq ||Qz_i - Qz_j||^2$, and $\langle z_i, z_j \rangle \neq \langle Qz_i, Qz_j \rangle$. This prevents us from using the properties of XU directly. In particular, if $Q \in \mathbb{O}(n)$, then we could use the fact that $\langle z_i, z_j \rangle = \langle Qz_i, Qz_j \rangle = 0$ if vertices i and j are in different communities.

The explicit form of XU represents points in $\mathbb{R}^{K^2}$ such that points within each community lie on K-dimensional orthogonal subspaces. Multiplication by $Q \in \mathbb{O}(p,q)$ removes the orthogonality property but retains the property that each community is represented by a K-dimensional subspace. Therefore, the ASE of P results in subspaces that correspond to each community, suggesting the use of SSC. In this paper, we will use a different and leave the properties of SSC on the ASE of A as future work.

Theorem 3. Let $P = VDV^\top$ be the spectral decomposition of the edge probability matrix. Let $B = VV^\top$. Then $B_{ij} = 0$ if vertices i and j are from different communities.

Theorem 3 provides perfect community detection given P. Letting $|B|$ be the affinity matrix for graph G, G is partitioned into K disjoint subgraphs that correspond to each community. Using A introduces some error in the ASE, resulting in points that lie near subspaces but not exactly on them. Using results from [13], we can show that as n grows, the ASE of A approaches the ASE of P and $B_{ij} \to 0$ for i and j in different clusters.

The ASE of A approaches latent positions that form P as the number of vertices n increases. More precisely, let each $\lambda_i^{(kl)} \sim \mathcal{F}^{(kl)}$ for $i = 1, .., n$ and $k, l = 1, ..., K$. Then the corresponding latent positions $XU \sim \mathcal{G}_K$ for some related joint distribution with K underlying communities $\mathcal{G}_K$. Denote Z_n as a sample of size n from $\mathcal{G}_K$ and adjacency matrix A_n as one draw from edge probability matrix $P_n = Z_n I_{pq} Z_n^\top$. Let $\hat{Z}_n$ be the adjacency embedding of A_n with rows $(\hat{z}_i^{(n)})^\top$. Then by Rubin-Delanchy et al. [13],

$$\max_{i \in \{1,...,n\}} ||Q_n \hat{z}_i^{(n)} - z_i^{(n)}|| = O_P\left(\frac{(\log n)^c}{n^{1/2}}\right) \tag{4}$$

for some $c > 0$ and sequence of $Q_n \in \mathbb{O}(p,q)$.

Theorem 4. Let $\hat{V}^{(n)} \in \mathbb{R}^{n \times K^2}$ be the matrix of K^2 eigenvectors of A_n corresponding to the $K(K+1)/2$ most positive eigenvalues and $K(K-1)/2$ most negative eigenvalues with rows $(\hat{v}_i^{(n)})^\top$. Let (i,j) correspond to pairs belonging to different communities. Let $n\rho_n = \omega((\log n)^{4c}))$ as described in Rubin-Delanchy et al. [13]. Then for some $c > 0$,

$$\max_{i,j} ||(\hat{v}_i^{(n)})^\top \hat{v}_j^{(n)}|| = O_P\left(\frac{(\log n)^c}{n\sqrt{\rho_n}}\right) \tag{5}$$

Thus, as $n \to \infty$, the partitioning of $\hat{B}^{(n)}$ results in perfect community detection. This leads

to the Orthogonal Spectral Clustering (OSC) algorithm (Alg. 1).

Algorithm 1: Orthogonal Spectral Clustering.

Data: Adjacency matrix A, number of communities K
Result: Community assignments $1, ..., K$

Compute the eigenvectors of A that correspond to the $K(K+1)/2$ most positive eigenvalues and $K(K-1)/2$ most negative eigenvalues. Construct V using these eigenvectors as its columns.
Compute $B = |VV^\top|$, applying $|\cdot|$ entry-wise.
Construct graph G using B as its similarity matrix.
Partition G into K disconnected subgraphs (e.g., using edge thresholding or spectral clustering).
Map each partition to the community labels $1, ..., K$.

Theorems 2, 3, and 4 also provide a very natural path toward using SSC for community detection for the PABM. We established in Theorem 2 that an ASE of the edge probability matrix P can be constructed such that the communities lie on mutually orthogonal subspaces, and this property can be recovered from the eigenvectors of P. Then Theorems 3 and 4 show that this property holds for the adjacency matrix A drawn from P as $n \to \infty$.

Algorithm 2: Sparse Subspace Clustering using LASSO [19].

Data: Adjacency matrix A, number of communities K, hyperparameter λ
Result: Community assignments $1, ..., K$

Find Z, the ASE of A using the $K(K+1)/2$ most positive and the $K(K-1)/2$ most negative eigenvalues and eigenvectors of A.
for $i = 1, ..., n$ **do**
  Assign $z_i^\top$ as the i^{th} row of Z. Assign $Z_{-i} = \begin{bmatrix} z_1 & \cdots & z_{i-1} & z_{i+1} & \cdots & z_n \end{bmatrix}^\top$.
  Solve the LASSO problem $c_i = \arg\min_{\beta_i} \frac{1}{2}||z_i - Z_{-i}\beta_i||_2^2 + \lambda||\beta_i||_1$.
  Assign $\tilde{c}_i = \begin{bmatrix} c_i^{(1)} & \cdots & c_i^{(i-1)} & 0 & c_i^{(i)} & \cdots & c_i^{(n-1)} \end{bmatrix}^\top$ such that the superscript is the index of $\tilde{c}_i$.
end
Assign $C = \begin{bmatrix} \tilde{c}_1 & \cdots & \tilde{c}_n \end{bmatrix}$.
Compute the affinity matrix $B = |C| + |C^\top|$.
Construct graph G using B as its similarity matrix.
Partition G into K disconnected subgraphs (e.g., using edge thresholding or spectral clustering).
Map each partition to the community labels $1, ..., K$.

Theorem 5. Let P_n describe the edge probability matrix of the PABM with n vertices, and let $A_n \sim \text{Bernoulli}(P_n)$. Let Z_n be the ASE of P_n, and let $Z_n^{(k)}$ correspond to the embedding points that correspond to the k^{th} community. Let $\hat{Z}_n$ be the ASE of A_n and $\hat{Z}_n^{(k)}$ be the corresponding k^{th} block. Let $\hat{B}_n$ be the affinity matrix in the SSC algorithm with input $\hat{Z}_n$ such that $\hat{B}_n^{(ij)}$ is the ij^{th} element of $\hat{B}_n$. If $r(Z_n^{(k)})$, the inradius of $Z_n^{(k)}$ (as defined by Soltanolkotabi and Candés [16]), is positive for each k, then $\hat{B}_n^{(ij)} \xrightarrow{p} 0 \ \forall i, j$ in different

communities.

3.3 Parameter estimation

For any edge probability matrix P for the PABM such that the rows and columns are organized by community, the kl^{th} block is an outer product of two vectors, i.e., $P^{(kl)} = \lambda^{(kl)}(\lambda^{(lk)})^\top$. Therefore, given $P^{(kl)}$, $\lambda^{(kl)}$ and $\lambda^{(lk)}$ are solvable exactly (up to multiplication by -1) using singular value decomposition. More specifically, let $P^{(kl)} = \sigma^2 uv^\top$ be the singular value decomposition of $P^{(kl)}$. $u \in \mathbb{R}^{n_k}$ and $v \in \mathbb{R}^{n_l}$ are vectors and $\sigma^2 > 0$ is a scalar. Then $\lambda^{(kl)} = \pm\sigma u$ and $\lambda^{(lk)} = \pm\sigma v$. Given the adjacency matrix A instead of edge probability matrix P, we can simply use plug-in estimators (algorithm 3), which converge to the true parameters (Theorem 6).

Algorithm 3: PABM parameter estimation using the adjacency matrix.

Data: Adjacency matrix A, community assignments $1, ..., K$
Result: PABM parameter estimates $\{\hat{\lambda}^{(kl)}\}_K$.

Arrange the rows and columns of A by community such that each $A^{(kl)}$ block consists of estimated edge probabilities between communities k and l.
for $k, l = 1, ..., K,\ k \leq l$ **do**
    Compute $A^{(kl)} = U\Sigma V^\top$, the SVD of the kl^{th} block.
    Assign $u^{(kl)}$ and $v^{(kl)}$ as the first columns of U and V. Assign $(\sigma^{(kl)})^2 \leftarrow \Sigma_{11}$.
    Assign $\hat{\lambda}^{(kl)} \leftarrow \pm\sigma^{(kl)}u^{(kl)}$ and $\hat{\lambda}^{(lk)} \leftarrow \pm\sigma^{(kl)}v^{(kl)}$.
end

Theorem 6. Under regularity and sparsity assumptions, and under the further assumption that K is fixed and community labels are known,

$$\max_{k,l\in\{1,...,K\}} ||\hat{\lambda}^{(kl)} - \lambda^{(kl)}|| = O_P\left(\frac{(\log n_k)^c}{\sqrt{n_k}}\right) \tag{6}$$

4 Simulated Examples

For each simulation, community labels are drawn from a multinomial distribution, the popularity vectors $\{\lambda^{(kl)}\}_K$ are drawn from two types of joint distributions depending on whether $k = l$, the edge probability matrix P is constructed using the popularity vectors, and finally an unweighted and undirected adjacency matrix A is drawn from P. OSC is then used for community detection, and this method is compared against SSC [12] [17] and MM [4] [15]. True community labels are used with Algorithm 3 to estimate the popularity vectors $\{\lambda^{(kl)}\}_K$, and this method is then compared against an MLE-based estimator described in Noroozi et al. [12] and Sengupta and Chen [15].

Modularity Maximization is NP-hard, so Sengupta and Chen [15] used the Extreme Points (EP) algorithm [9], which is $O(n^{K-1})$. For these simulations, the EP algorithm was used for $K = 2$, and for $K > 2$, the Louvain algorithm [3] was used instead.

Two implementations of SSC are shown here. The first method, denoted as SSC-A, treats the columns of the adjacency matrix A as points in $\mathbb{R}^n$, as described in Noroozi et al. [12]. The second method, denoted as SSC-ASE, first embeds A and then performs SSC on the embedding, as described in algorithm 2. The sparsity parameter λ was chosen via a preliminary cross-validation experiment. For the final clustering step, a Gaussian Mixture Model was fit on the normalized Lapacian eigenmap of the affinity matrix B.

4.1 Balanced communities

In each simulation, community labels $z_1, ..., z_n$ were drawn from a multinomial distribution with mixture parameters $\{\alpha_1, ..., \alpha_K\}$, then $\{\lambda^{(kl)}\}_K$ according to the drawn community labels, P was constructed using the drawn $\{\lambda^{(kl)}\}_K$, and A was drawn from P by $A_{ij} \overset{indep}{\sim} Bernoulli(P_{ij})$. Each simulation has a unique edge probability matrix P.

For these examples, we set the following parameters:

- Number of vertices $n = 128, 256, 512, 1024, 2048, 4096$
- Number of underlying communities $K = 2, 3, 4$
- Mixture parameters $\alpha_k = 1/K$ for $k = 1, ..., K$, (i.e., each community label has an equal probability of being drawn)
- Community labels $z_k \overset{\text{iid}}{\sim} Multinomial(\alpha_1, ..., \alpha_K)$
- Within-group popularities $\lambda^{(kk)} \overset{\text{iid}}{\sim} Beta(2, 1)$
- Between-group popularities $\lambda^{(kl)} \overset{\text{iid}}{\sim} Beta(1, 2)$ for $k \neq l$

50 simulations were performed for each (n, K) pair.

Fig. 1 show that for large n, OSC results in a misclustering rate of 0. Sparse subspace clustering produces similar results for $K > 2$.

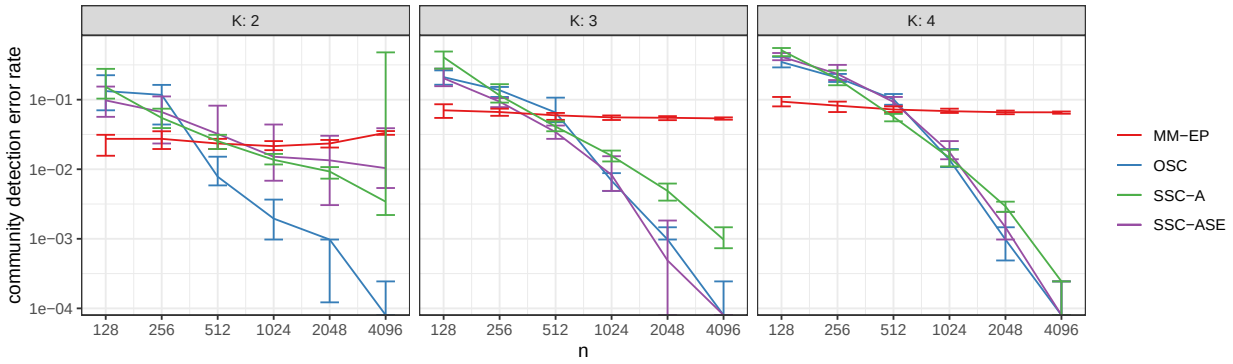

Figure 1: IQR of clustering error using OSC (blue) compared against SSC on the ASE of A (purple), MM (red), and SSC on the adjacency matrix (green). Communities are approximately balanced. Simulations were repeated 50 times for each sample size.

Theorem 4 implies that OSC will result in not just in the error rate converging to 0 but the error *count* as well. We explore this in Fig. 2.

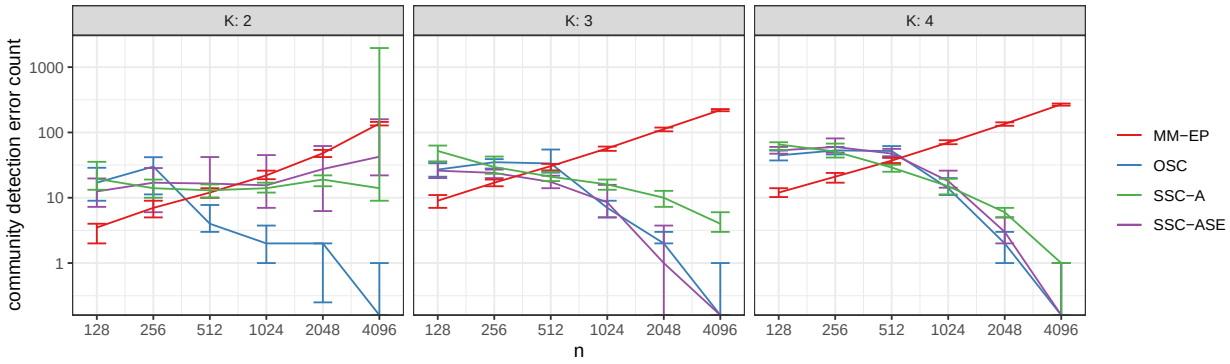

Figure 2: IQR of counts of misclustered vertices using OSC (blue) compared against SSC on the ASE of A (purple), MM (red), and SSC on the adjacency matrix (green). Communities are approximately balanced. Simulations were repeated 50 times for each sample size.

Given ground truth community labels, Algorithm 3 and the MLE-based plug-in estimators [15] [12] perform similarly, with root mean square error decaying at rate approximately $n^{-1/2}$.

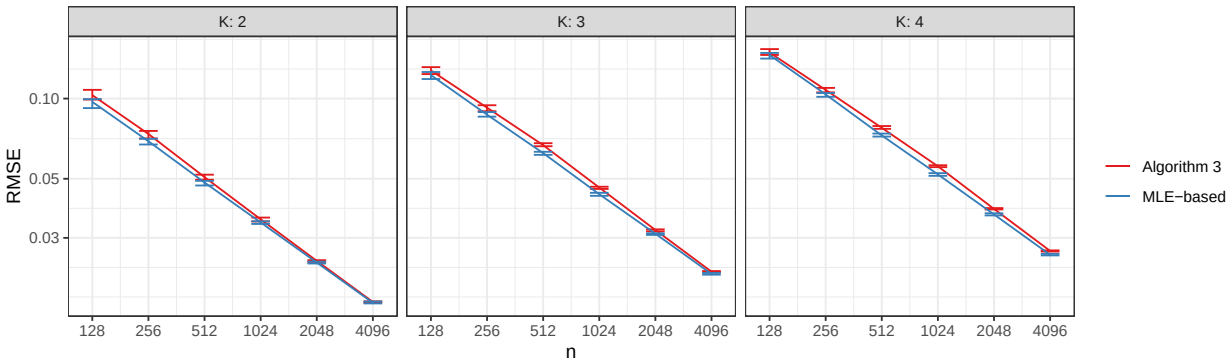

Figure 3: Median and IQR RMSE from Algorithm 3 (red) compared against an MLE-based method (blue). Simulations were repeated 50 times for each sample size.

4.2 Imbalanced communities

Simulations performed in this section are similar to those in the previous section with the exception of the mixture parameters $\{\alpha_1, ..., \alpha_K\}$ used to draw community labels from the multinomial distribution. For these examples, we set the following parameters:

- Number of vertices $n = 128, 256, 512, 1024, 2048, 4096$
- Number of underlying communities $K = 2, 3, 4$
- Mixture parameters $\alpha_k = \frac{k^{-1}}{\sum_{l=1}^{K} l^{-1}}$ for $k = 1, ..., K$
- Community labels $z_k \overset{\text{iid}}{\sim} Multinomial(\alpha_1, ..., \alpha_K)$
- Within-group popularities $\lambda^{(kk)} \overset{\text{iid}}{\sim} Beta(2, 1)$
- Between-group popularities $\lambda^{(kl)} \overset{\text{iid}}{\sim} Beta(1, 2)$ for $k \neq l$

50 simulations were performed for each (n, K) pair.

Fig. 4 and 5 show similar results as in the balanced communities case, with both OSC and SSC resulting in no misclustered vertices for a sufficiently large sample size. However, Fig. 6 suggests that while Algorithm 3 retains $\sqrt{n}$ efficiency, the MLE-based plug-in estimator is less efficient for this setup.

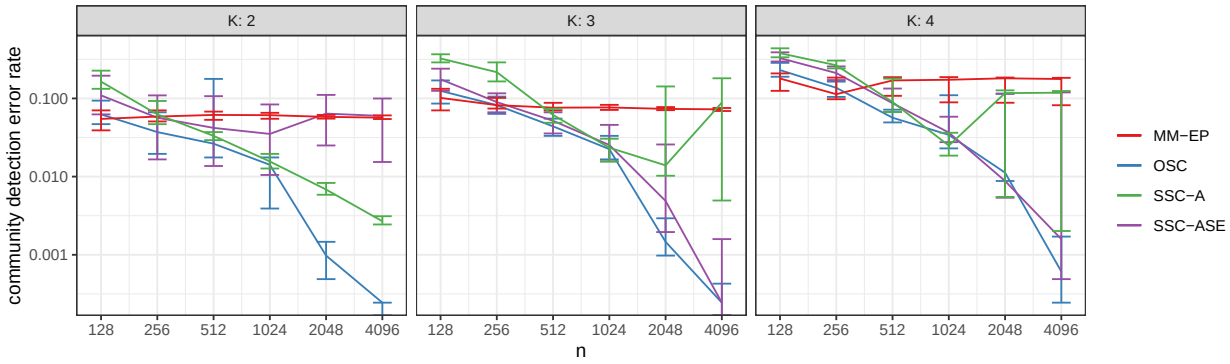

Figure 4: IQR of clustering error using OSC (blue) compared against SSC on the ASE of A (purple), MM (red), and SSC on the adjacency matrix (green). Communities are imbalanced. Simulations were repeated 50 times for each sample size.

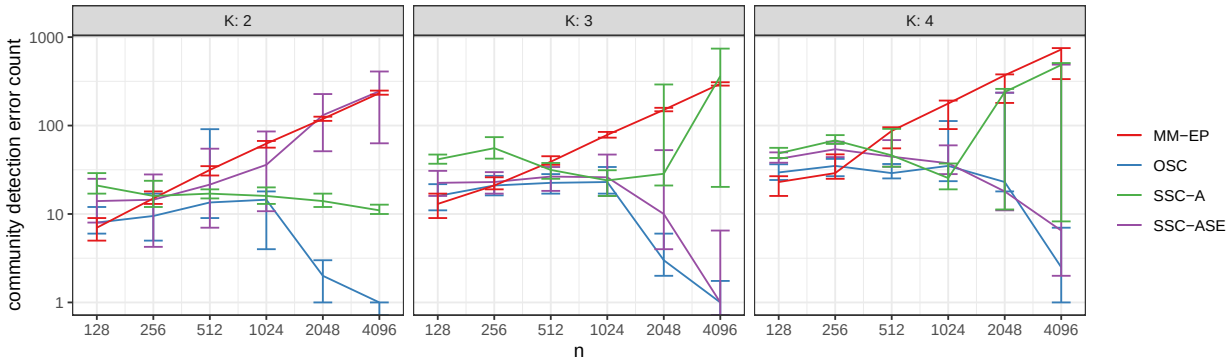

Figure 5: IQR of counts of misclustered vertices using OSC (blue) compared against SSC on the ASE of A (purple), MM (red), and SSC on the adjacency matrix (green). Communities are imbalanced. Simulations were repeated 50 times for each sample size.

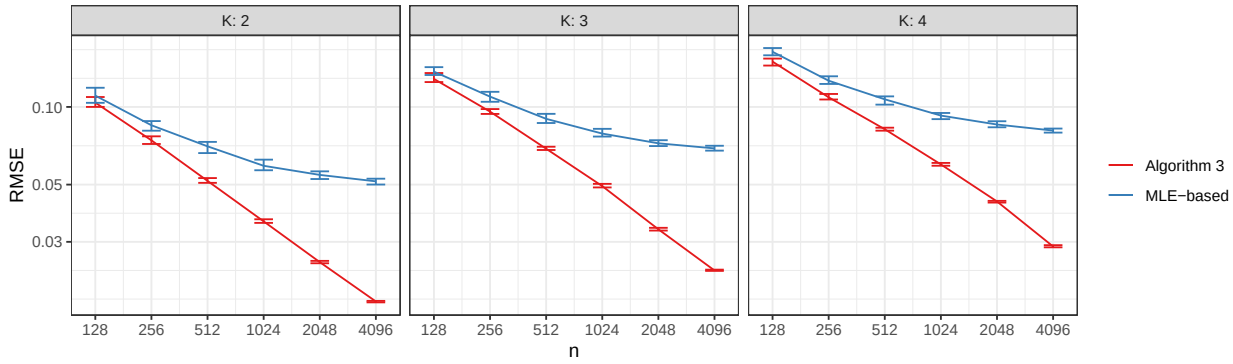

Figure 6: Median and IQR RMSE from Algorithm 3 (red) compared against an MLE-based method (blue). Simulations were repeated 50 times for each sample size.

4.3 Additional experiments

Using the same set of parameters for generating P and A as in the balanced communities examples for $K = 2$, we generated one instance of A for each n and constructed B according to Algorithm 3 to verify that as $n \to \infty$, $(\hat{v}_i)^\top \hat{v}_j \xrightarrow{P} 0$ for i, j in different clusters. Furthermore, the distribution of these inner products should be approximately normal.

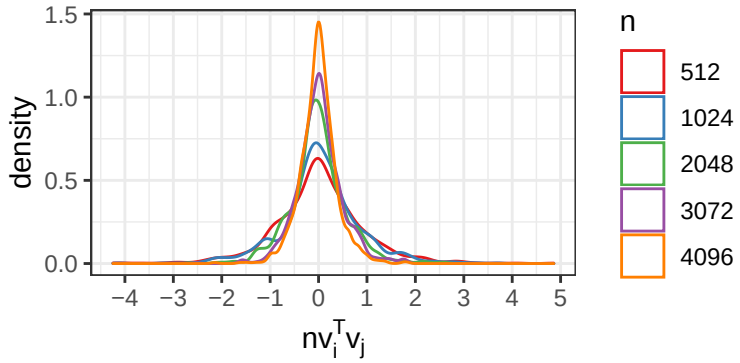

Figure 7: Between-cluster inner products of the eigenvectors of A for varying sample sizes.

5 Real data examples

In the first real data example, we applied OSC to the Leeds Butterfly dataset [18] consisting of visual similarity measurements among 832 butterflies across 10 species. The graph was modified to match the example from Noroozi et al. [12]: Only the 4 most frequent species were considered, and the similarities were discretized to $\{0, 1\}$ via thresholding. Fig. 8 shows a sorted adjacency matrix sorted by the resultant clustering.

Comparing against the ground truth species labels, OSC achieves an accuracy of 63% and an adjusted Rand index of 73%. In comparison, Noroozi et al. [12] achieved an adjusted Rand index of 73% using sparse subspace clustering on the same dataset.

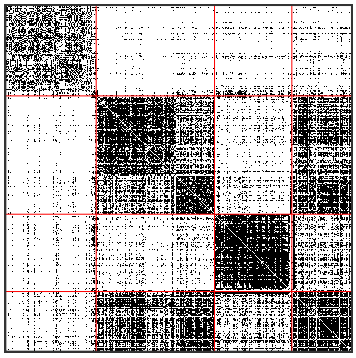

Figure 8: Adjacency matrix of the Leeds Butterfly dataset after sorting by the clustering outputted by OSC.

In the second example, we applied OSC to the British MPs Twitter network [6], the Political Blogs network [1], and the DBLP network [5] [7]. For this data analysis, we subsetted the data as described by Sengupta and Chen [15] for their analysis of the same networks. Our methods underperformed compared to modularity maximization, although performance is comparable. In addition, OSC's runtime is much lower than that of modularity maximization.

Table 1: Community detection error rates for modularity maximization, sparse subspace clustering, and OSC.

Network	MM	SSC-ASE	OSC
British MPs	0.003	0.018	0.009
Political blogs	0.050	0.196	0.062
DBLP	0.028	0.087	0.059

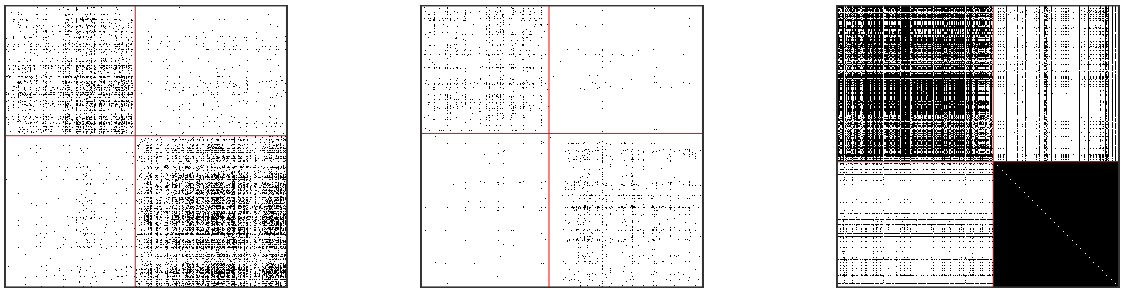

Figure 9: Adjacency matrices of (from left to right) the British MPs, Political Blogs, and DBLP networks after sorting by the clustering outputted by OSC.

In the third example, we consider the Karantaka villages data studied by Banerjee et al. [2]. For this example, we chose the `visitgo` networks from villages 12, 31, and 46 at the

household level. The label of interest is the religious affiliation. The networks were truncated to religions "1" and "2", and vertices of degree 0 were removed.

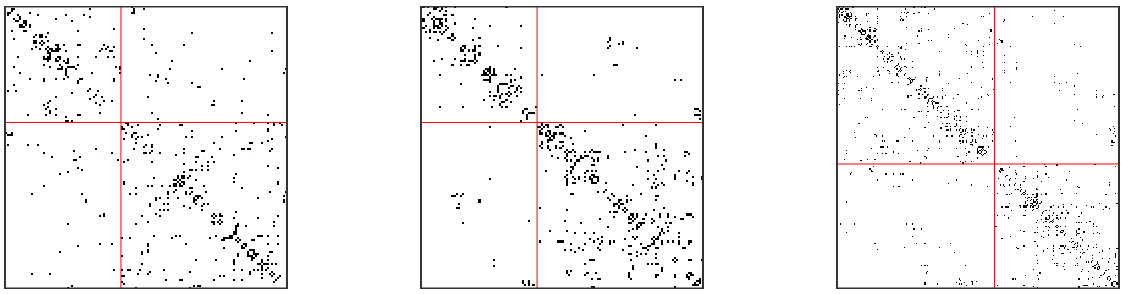

Figure 10: Adjacency matrix of the Karnataka villages data, arranged by the clustering produced by OSC (left). The villages studied here are, from left to right, 12, 31, and 46.

Table 2: Community detection error rates for identifying household religion.

Network	MM	SSC-ASE	OSC
Village 12	0.270	0.291	0.227
Village 31	0.125	0.169	0.110
Village 46	0.052	0.208	0.078

6 Discussion

7 Proofs

Proof of Theorem 1. This is given by straightforward matrix multiplication. It suffices to show that

$$XUI_{3,1}U^\top X^\top = \begin{bmatrix} \lambda^{(11)}(\lambda^{(11)})^\top & \lambda^{(12)}(\lambda^{(21)})^\top \\ \lambda^{(21)}(\lambda^{(12)})^\top & \lambda^{(22)}(\lambda^{(22)})^\top \end{bmatrix}$$

Remark. While we can just perform the matrix multiplication to show the equivalence, it is more illustrative to look at a few intermediate steps. Note that the product of the three inner matrices results in a permutation matrix with fixed points at positions 1 and 4 and a cycle of order 2 swappoing positions 2 and 3:

$$UI_{3,1}U^\top = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \Pi$$

Since U is orthonormal and $I_{3,1}$ is diagonal, $\Pi = UI_{3,1}U^\top$ is a spectral decomposition of this permutation matrix. Note that the two fixed points result in eigenvalues of $+1$ with corresponding eigenvectors e_i where $i = 1, 4$ corresponding to the locations of the fixed points, and the cycle of order two results in two eigenvalues ± 1 with corresponding eigenvectors $(e_i \pm e_j)/\sqrt{2}$ where $i = 2, j = 3$, pair that is swapped.

Proof of Theorem 2. Let

$$\Lambda^{(k)} = \begin{bmatrix} \lambda^{(k,1)} & \cdots & \lambda^{(k,K)} \end{bmatrix} \in \mathbb{R}^{n_k \times K}$$

$$X = \text{blockdiag}(\Lambda^{(1)}, ..., \Lambda^{(K)}) \in \mathbb{R}^{n \times K^2} \tag{7}$$

$$L^{(k)} = \text{blockdiag}(\lambda^{(1k)}, ..., \lambda^{(Kk)}) \in \mathbb{R}^{n \times K}$$

$$Y = \begin{bmatrix} L^{(1)} & \cdots & L^{(K)} \end{bmatrix} \in \mathbb{R}^{n \times K^2}$$

Then $P = XY^\top$.

Similar to the $K = 2$ case, we have $Y = X\Pi$ for a permutation matrix Π, resulting in $P = X\Pi X^\top$.
The permutation described by Π has K fixed points, which correspond to K eigenvalues equal to 1 with corresponding eigenvectors e_k where $k = r(K + 1) + 1$ for $r = 0, ..., K - 1$. It also has $\binom{K}{2} = K(K - 1)/2$ cycles of order 2. Each cycle corresponds to a pair of eigenvalues $+1$ and -1 and a pair of eigenvectors $(e_s + e_t)/\sqrt{2}$ and $(e_s - e_t)/\sqrt{2}$.

Then Π has $K(K + 1)/2$ eigenvalues equal to 1 and $K(K - 1)/2$ eigenvalues equal to -1. Π has the decomposed form

$$\Pi = UI_{K(K+1)/2,K(K-1)/2}U^\top \tag{8}$$

The edge probability matrix then can be written as:

$$P = XUI_{p,q}(XU)^\top \tag{9}$$

$$p = K(K + 1)/2 \tag{10}$$

$$q = K(K - 1)/2 \tag{11}$$

and we can describe the PABM with K communities as a GRDPG with latent positions XU with signature $\Big(K(K + 1)/2, K(K - 1)/2\Big)$.

Lemma 3.1. Let $P = VDV^\top$ be the spectral decomposition of the edge probability matrix for a PABM. Then $VV^\top = X(X^\top X)^{-1}X^\top$ where X is defined as in (7).

Proof of Lemma 3.1. By Theorem 2, $P = XUI_{p,q}U^\top X^\top$, where X is defined as in (7) and p and q are defined as in equations (10) and (11). Alternatively, the spectral decomposition can be written as $P = VDV^\top = V|D|^{1/2}I_{p,q}|D|^{1/2}V^\top$ for the same (p, q) and $|\cdot|^{1/2}$ is applied entry-wise. Thus for some $Q \in \mathbb{O}(p, q)$,

$$XUQ = V|D|^{1/2}$$

Therefore, using the fact that $UU^\top = I$ and $V^\top V = I$,

$$(V|D|^{1/2})((V|D|^{1/2})^\top(V|D|^{1/2}))^{-1}(V|D|^{1/2})^\top = (XUQ)((XUQ)^\top(XUQ))^{-1}(XUQ)^\top$$

The righthand side becomes

$$\begin{aligned}(XUQ)((XUQ)^\top(XUQ))^{-1}(XUQ)^\top &= XUQQ^{-1}U^\top(X^\top X)^{-1}U(Q^\top)^{-1}Q^\top U^\top X^\top \\ &= XUU^\top(X^\top X)^{-1}UU^\top X^\top \\ &= X(X^\top X)^{-1}X^\top\end{aligned}$$

The lefthand side becomes:

$$\begin{aligned}(V|D|^{1/2})((V|D|^{1/2})^\top(V|D|^{1/2}))^{-1}(V|D|^{1/2})^\top &= V|D|^{1/2}|D|^{-1/2}(V^\top V)^{-1}|D|^{-1/2}|D|^{1/2}V^\top \\ &= VV^\top\end{aligned}$$

Proof of Theorem 3. By Lemma 3.1, $VV^\top = X(X^\top X)^{-1}X^\top$ where X is defined as in Theorem 2. Since X is block diagonal with each block corresponding to one community, $X(X^\top X)^{-1}X^\top$ is also a block diagonal matrix with each block corresponding to a community and zeros elsewhere. Therefore, if vertices i and j belong to different communities, then the ij^{th} element of $X(X^\top X)^{-1}X^\top = VV^\top = B$ is 0.

Lemma 5.1. Let $P_n = V_n\Lambda_n V_n^\top$ be defined as in Lemma 5.1. Let $A_n \sim \text{Bernoulli}(P_n)$ be the adjacency matrix drawn from P_n, and let $A_n = \hat{V}_n\hat{\Lambda}_n\hat{V}_n^\top$ be its approximate spectral decomposition using the $K(K+1)/2$ most positive and $K(K-1)/2$ most negative eigenvalues and their corresponding eigenvectors. Without loss of generality, order the rows of V_n and $\hat{V}_n$ by community, and let $V_n^{(k)}$ and $\hat{V}_n^{(k)}$ be the $n_k \times K^2$ matrix of the rows of the k^{th} community in V_n and $\hat{V}_n$ respectively. Let $r_n^{(k)} = r(V_n^{(k)})$ and $\hat{r}_n^{(k)} = r(\hat{V}_n^{(k)})$ be the inradius of points in V_n and $\hat{V}_n$ as defined by Soltanolkotabi and Candés [16] and Wang and Xu [19]. Then

$$\hat{r}_n^{(k)} \xrightarrow{p} r_n^{(k)} \tag{12}$$

Proof of Lemma 5.1. This falls directly out of Theorem 5 of Rubin-Delanchy et al. [13]. Since $\hat{V}_n^{(k)} \xrightarrow{p} V_n^{(k)}$ and $r(\cdot)$ is continuous, $r(\hat{V}_n^{(k)}) \xrightarrow{p} r(V_n^{(k)})$.

Lemma 5.2. Let $\mu_n^{(k)} = \mu(V_n^{(k)})$ be the subspace incoherence of $V_n^{(k)}$ as defined by Soltanolkotabi and Candés [16]. Then $\mu_n^{(k)} = 0$ with probability 1 $\forall n, k$.

Proof of Lemma 5.2. Since $V_n^{(k)}$ is orthogonal to $V_n^{(l)}$ $\forall k \neq l$ (Lemma 3.1), this falls directly out of Theorem 2.8 of Soltanolkotabi and Candés [16].

Lemma 5.3. Let $\hat{\mu}_n^{(k)} = \mu(\hat{V}_n^{(k)})$ using the extended definition of $\mu(\cdot)$ from Wang and Xu [19]. Then

$$\hat{\mu}_n^{(k)} \xrightarrow{p} 0 \tag{13}$$

Proof of Lemma 5.3. This falls directly out of Theorem 5 of Rubin-Delanchy et al. [13]. Since $\hat{V}_n^{(k)} \xrightarrow{p} V_n^{(k)}$ and $\mu(\cdot)$ is continuous, $\mu(\hat{V}_n^{(k)}) \xrightarrow{p} \mu(V_n^{(k)}) = 0$.

Lemma 5.4. Let $(v_n^{(i)})^\top$ and $(\hat{v}_n^{(i)})^\top$ be the rows of V_n and $\hat{V}_n$ respectively. By Rubin-Delanchy et al. [13],

$$\delta_n = \max_i ||\hat{v}_n^{(i)} - v_n^{(i)}|| \xrightarrow{p} 0 \tag{14}$$

Proof of Theorem 5. For ease of proving, we will consider V_n and $\hat{V}_n$ instead of Z_n and $\hat{Z}_n$.

By (12) and (13),

$$r(\hat{V}_n^{(k)}) - \mu(\hat{V}_n^{(k)}) \xrightarrow{p} r_n^{(k)} - \mu_n^{(k)} = r_n^{(k)} > 0 \tag{15}$$

for each $k = 1, ..., K$.

Then by Theorem 6 of Wang and Xu [19], (14) and (15) imply that $\forall \epsilon > 0$, $\exists N > 0$ such that $\forall n > N$, Pr(SSC detection property holds for $\hat{V}_n$) $= 1 - \epsilon$.

Proof of Theorem 6. Let P and A be organized by community such that the elements of blocks $P^{(kl)}$ and $A^{(kl)}$ correspond to the edges between communities k and l.

Case $k = l$. $P^{(kk)}$ and $A^{(kk)}$ represent within-community edge probabilities and edges for community k.
By definition, $P^{(kk)} = \lambda^{(kk)}(\lambda^{(kk)})^\top$. This implies that the signular value decomposition $P^{(kk)} = \sigma_{kk}^2 u^{(kk)}(u^{(kk)})^\top$ has one singular value and one pair of singular vectors ($P^{(kk)}$ is symmetric, so the left and right singular vectors are identical). Then $\lambda^{(kk)} = \sigma_{kk} u^{(kk)}$.
Let $\hat{U}^{(kk)}\hat{\Sigma}^{(kk)}(\hat{U}^{(kk)})^\top$ be the singular value decomposition of $A^{(kk)}$, and let $\hat{\sigma}_{kk}^2 \hat{u}^{(kk)}(\hat{u}^{(kk)})^\top$ be its one-dimensional approximation. Define $\hat{\lambda}^{(kk)} = \hat{\sigma}_{kk}\hat{u}^{(kk)}$. Then $\hat{\lambda}^{(kk)}$ is the adjacency spectral embedding approximation of $\lambda^{(kk)}$.
Then by Theorem 5 from Rubin-Delanchy et al. [13], the adjacency spectral embedding $\hat{\lambda}^{(kk)}$ approximates $\lambda^{(kk)}$ at rate $\frac{(\log n_k)^c}{\sqrt{n_k}}$.

Case $k \neq l$. $P^{(kl)}$ and $A^{(kl)}$ represent edge probabilities and edges between communities k and l. Note that $P^{(kl)} = (P^{(lk)})^\top$.
By definition, $P^{(kl)} = \lambda^{(kl)}(\lambda^{(lk)})^\top$. As in the $k = l$ case, we note that the singular value decomposition $P^{(kl)} = \sigma_{kl}^2 u^{(kl)}(v^{(kl)})^\top$ is one-dimensional and $\lambda^{(kl)} = \sigma_{kl} u^{(kl)}$. (We can also note that the SVD of $P^{(lk)} = \sigma_{kl}^2 v^{(kl)}(u^{(kl)})^\top$, i.e., $\sigma_{kl} = \sigma_{lk}$, $u^{(kl)} = v^{(lk)}$, and $v^{(kl)} = u^{(lk)}$.)
Now consider the Hermitian dilation

$$M^{(kl)} = 2\begin{bmatrix} 0 & P^{(kl)} \\ P^{(lk)} & 0 \end{bmatrix}$$

which is a symmetric $(n_k + n_l) \times (n_k + n_l)$ matrix. It can be shown that the spectral decomposition of $M^{(kl)}$ is

$$M^{(kl)} = \begin{bmatrix} u^{(kl)} & -u^{(kl)} \\ v^{(kl)} & v^{(kl)} \end{bmatrix} \times \begin{bmatrix} \sigma_{kl}^2 & 0 \\ 0 & -\sigma_{kl}^2 \end{bmatrix} \times \begin{bmatrix} u^{(kl)} & -u^{(kl)} \\ v^{(kl)} & v^{(kl)} \end{bmatrix}^\top$$

Thus treating $M^{(kl)}$ as the edge probability matrix of a GRDPG, we have latent positions in $\mathbb{R}^2$ given by

$$\begin{bmatrix} \sigma_{kl} u^{(kl)} & \sigma_{kl} u^{(kl)} \\ \sigma_{kl} v^{(kl)} & -\sigma_{kl} v^{(kl)} \end{bmatrix} = \begin{bmatrix} \lambda^{(kl)} & \lambda^{(kl)} \\ \lambda^{(lk)} & -\lambda^{(lk)} \end{bmatrix}$$

Now consider

$$\hat{M}^{(kl)} = \begin{bmatrix} 0 & A^{(kl)} \\ A^{(lk)} & 0 \end{bmatrix}$$

Then $\hat{M}^{(kl)} = M^{(kl)} + E'$ where

$$E' = \begin{bmatrix} 0 & E \\ E^\top & 0 \end{bmatrix}$$

and E is the $n_k \times n_l$ matrix of independent noise (to generate the Bernoulli entries in $A^{(kl)}$. Then $\hat{M}^{(kl)}$ is an adjacency matrix drawn from $M^{(kl)}$, so its adjacency spectral embedding, given by

$$\begin{bmatrix} \hat{\lambda}^{(kl)} & \hat{\lambda}^{(kl)} \\ \hat{\lambda}^{(lk)} & -\hat{\lambda}^{(lk)} \end{bmatrix}$$

where each $\hat{\lambda}^{(kl)}$ is defined as in Algorithm 3, approximates the latent positions of $M^{(kl)}$ up to indefinite orthogonal transformation by the rate given in Theorem 5 of Rubin-Delanchy et al. [13].
In this case, the indefinite orthogonal transformation W_* in the GRDPG result [13] is of

the form $U^\top \hat{U}$. The eigenvalues of M are distinct since the signature for this GRDPG is $(1, 1)$, and $U^\top \hat{U}$ is block diagonal, resulting in $W_* \xrightarrow{P} I$. Therefore, the adjacency spectral embedding of $\hat{M}^{(kl)}$ is a direct estimation of the specific latent positions outlined for $M^{(kl)}$, up to sign flip.

References

[1] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, page 36–43, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595932151. doi: 10.1145/1134271.1134277. URL https://doi.org/10.1145/1134271.1134277.

[2] Abhijit Banerjee, Arun G. Chandrasekhar, Esther Duflo, and Matthew O. Jackson. The Diffusion of Microfinance, 2013. URL https://doi.org/10.7910/DVN/U3BIHX.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, Oct 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/p10008. URL http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

[4] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL https://igraph.org.

[5] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 585–593. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3855-graph-based-consensus-maximization-among-multiple-supervised-and-unsupervised-models.pdf.

[6] Derek Greene and Pádraig Cunningham. Producing a unified graph representation from multiple social network views. *CoRR*, abs/1301.5809, 2013. URL http://arxiv.org/abs/1301.5809.

[7] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. Graph regularized transductive classification on heterogeneous information networks. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 570–586, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15880-3.

[8] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), Jan 2011. ISSN 1550-2376. doi: 10.1103/physreve.83.016107. URL http://dx.doi.org/10.1103/PhysRevE.83.016107.

[9] Can M. Le, Elizaveta Levina, and Roman Vershynin. Optimization via low-rank approximation for community detection in networks. *Ann. Statist.*, 44(1):373–400, 02 2016. doi: 10.1214/15-AOS1360. URL https://doi.org/10.1214/15-AOS1360.

[10] François Lorrain and Harrison C. White. Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1(1):49–80, 1971. doi: 10.1080/0022250X.1971.9989788. URL https://doi.org/10.1080/0022250X.1971.9989788.

[11] Vince Lyzinski, Daniel L. Sussman, Minh Tang, Avanti Athreya, and Carey E. Priebe. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electron. J. Statist.*, 8(2):2905–2922, 2014. doi: 10.1214/14-EJS978. URL https://doi.org/10.1214/14-EJS978.

[12] Majid Noroozi, Ramchandra Rimal, and Marianna Pensky. Estimation and Clustering in Popularity Adjusted Stochastic Block Model. *arXiv e-prints*, art. arXiv:1902.00431, February 2019.

[13] Patrick Rubin-Delanchy, Joshua Cape, Minh Tang, and Carey E. Priebe. A statistical interpretation of spectral embedding: the generalised random dot product graph, 2017.

[14] Patrick Rubin-Delanchy, Carey E. Priebe, and Minh Tang. Consistency of adjacency spectral embedding for the mixed membership stochastic blockmodel, 2017.

[15] Srijan Sengupta and Yuguo Chen. A block model for node popularity in networks with community structure. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 80(2):365–386, March 2018. ISSN 1369-7412. doi: 10.1111/rssb.12245.

[16] Mahdi Soltanolkotabi and Emmanuel J. Candés. A geometric analysis of subspace clustering with outliers. *Ann. Statist.*, 40(4):2195–2238, 08 2012. doi: 10.1214/12-AOS1034. URL https://doi.org/10.1214/12-AOS1034.

[17] Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *Ann. Statist.*, 42(2):669–699, 04 2014. doi: 10.1214/13-AOS1199. URL https://doi.org/10.1214/13-AOS1199.

[18] Bo Wang, Armin Pourshafeie, Marinka Zitnik, Junjie Zhu, Carlos D. Bustamante, Serafim Batzoglou, and Jure Leskovec. Network enhancement as a general method to denoise weighted biological networks. *Nature Communications*, 9(1), Aug 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-05469-x. URL http://dx.doi.org/10.1038/s41467-018-05469-x.

[19] Yu-Xiang Wang and Huan Xu. Noisy sparse subspace clustering. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 89–97, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/wang13.html.

[20] Stephen J. Young and Edward R. Scheinerman. Random dot product graph models for social networks. In Anthony Bonato and Fan R. K. Chung, editors, *Algorithms and Models for the Web-Graph*, pages 138–149, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-77004-6.