

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

## Abstract

In this paper, we connect two probabilistic models for graphs, the Popularity Adjusted Block Model (PABM) and the Generalized Random Dot Product Graph (GRDPG) and use properties established in this connection to aid in clustering and parameter estimation. In particular, we note that the PABM can be represented as latent positions such that points within the same cluster lie on a subspace, and the subspaces that represent each cluster 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 clustering and parameter estimation methods for the PABM.

## 1 Introduction

Statistical analysis on graphs and networks often involve the partitioning of a graph into clusters. This is often motivated by the belief that there is an underlying and unobserved community or cluster label for each vertex of the graph, and edges between pairs of vertices are determined by drawing from a probability distribution based on the communities assigned to each pair. One such probabilistic model is the Stochastic Block Model (SBM), which assumes that the edge probability from one vertex to another follows a Bernoulli distribution with fixed probabilities for each pair of community labels. The Popularity Adjusted Block Model (PABM) was then introduced by Sengupta and Chen [4] as a generalization of the SBM to address the heterogeneity of edge probabilities within and between communities while still maintaining the community structure.

The Random Dot Product Graph (RDPG) model [1] is another graph model with Bernoulli edge probabilities. Under this model, each vertex of the graph can be represented by a point in latent space such that the edge probability between any pair of vertices is given by their corresponding dot product in the latent space. The SBM is equivalent to a special case of the RDPG model in which all vertices in a given community share the same position in the latent space. It has also been shown that similar probabilistic graph models, (such as the Mixed Membership Stochastic Block Model [3]). An analogous property exists for the PABM not under the RDPG model but under the *Generalized* Random Dot Product Graph (GRDPG) model.

In this paper, we will only consider undirected graphs, that is the edge value from vertices

$i$  and  $j$  is equal to the edge in the opposite direction, from vertices  $j$  and  $i$ . Furthermore, we will only consider unweighted graphs with binary edge values. We will also assume that graphs are hollow, i.e., there are no self-loops.

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

### 2.1 The popularity adjusted block model (PABM) [2]

**Definition 1.** Let  $P \in [0, 1]^{n \times n}$  be an edge probability matrix for a set of  $n$  vertices,  $V$ . Each vertex has a community label  $1, \dots, K$ , 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  $l$ . 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.,  $A_{ij} \stackrel{\text{indep}}{\sim} \text{Bernoulli}(P_{ij})$  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  $\lambda^{(kl)}(\lambda^{(lk)})^\top$  for a set of 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  $A \sim \text{PABM}(\{\lambda^{(kl)}\}_K)$  to denote a random adjacency matrix  $A$  drawn from a PABM with parameters  $\lambda^{(kl)}$  consisting of  $K$  underlying clusters/communities.

### 2.2 The generalized random dot product graph (GRDPG) [3]

**Definition 2.** Let  $X \in \mathbb{R}^{n \times d}$  be latent positions of the vertices of a graph  $G$ .  $X$  consists of row vectors  $x_i^\top$ . Let  $A \in \{0, 1\}^{n \times n}$  be the corresponding adjacency matrix.

Fix  $p, q$  such that  $p + q = d$  and define  $I_{pq} = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}$ .

Then  $G = (V, E)$  is a generalized random dot product graph with signature  $(p, q)$  and latent positions  $X$  iff its random adjacency matrix can be described as  $A_{ij} \stackrel{\text{indep}}{\sim} \text{Bernoulli}(P_{ij})$  where  $P_{ij} = x_i^\top I_{pq} x_j$ .

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

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

## 2.3 Connecting the PABM to the GRDPG

### 2.3.1 Case where $K = 2$

**Theorem 1.** Let  $X = \begin{bmatrix} \lambda^{(11)} & \lambda^{(12)} & 0 & 0 \\ 0 & 0 & \lambda^{(21)} & \lambda^{(22)} \end{bmatrix}$ , and let  $U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix}$ , as

in Definition 1. Then  $A \sim \text{GRDPG}_{3,1}(XU)$  and  $A \sim \text{PABM}(\{(\lambda^{(kl)})\}_K)$  are equivalent.

*Proof.* Let  $X = \begin{bmatrix} \lambda^{(11)} & \lambda^{(12)} & 0 & 0 \\ 0 & 0 & \lambda^{(21)} & \lambda^{(22)} \end{bmatrix}$  and  $Y = \begin{bmatrix} \lambda^{(11)} & 0 & \lambda^{(12)} & 0 \\ 0 & \lambda^{(21)} & 0 & \lambda^{(22)} \end{bmatrix}$ . Then  $P = XY^\top$ .

We can note that  $Y = X\Pi$  where  $\Pi$  is the permutation matrix  $\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ . Therefore,

$$P = X\Pi X^\top.$$

Taking the spectral decomposition of  $\Pi = UDU^\top$ , we can see that  $P = (XU)D(XU)^\top$ . We can then denote  $\Sigma = |D|^{1/2}$ , the square root of the absolute values of the (diagonal) entries of  $D$  and obtain  $P = (XU\Sigma)I_{pq}(XU\Sigma)^\top$  where  $p$  and  $q$  correspond to the number of positive and negative eigenvalues of  $\Pi$ , respectively. Therefore, the PABM with  $K = 2$  is a special case of the GRDPG. We can however expand upon this a bit further.

The permutation described by  $\Pi$  has two fixed points and one cycle of order 2. The two fixed points are at positions 1 and 4, so  $\Pi$  has two eigenvalues equal to 1 and corresponding eigenvectors  $e_1$  and  $e_4$ . The cycle of order 2 switching positions 2 and 3 corresponds to eigenvalues 1 and  $-1$  with corresponding eigenvalues  $(e_2+e_3)/\sqrt{2}$  and  $(e_2-e_3)/\sqrt{2}$  respectively.

$$\text{Therefore, } D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = I_{3,1} \text{ and } U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Putting it all together, we get  $P = (XU)I_{3,1}(XU)^\top$ . Therefore, the PABM with  $K = 2$  is a GRDPG with  $p = 3$ ,  $q = 1$ ,  $d = K^2 = 4$ , and latent positions  $XU = \begin{bmatrix} \lambda^{(11)} & 0 & \lambda^{(12)}/\sqrt{2} & \lambda^{(12)}/\sqrt{2} \\ 0 & \lambda^{(22)} & \lambda^{(21)}/\sqrt{2} & -\lambda^{(21)}/\sqrt{2} \end{bmatrix}$ .

### 2.3.2 Generalization to $K > 2$

**Theorem 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  $A \sim \text{GRDPG}_{K(K+1)/2, K(K-1)/2}(XU)$  and  $A \sim \text{PABM}(\{(\lambda^{(kl)})\}_K)$  are equivalent.

*Proof* Let  $\Lambda^{(k)} = [\lambda^{(k,1)} \ \dots \ \lambda^{(k,K)}] \in \mathbb{R}^{n_k \times K}$ .

Let  $X$  be a block diagonal matrix  $X = \text{diag}(\Lambda^{(1)}, \dots, \Lambda^{(K)}) \in \mathbb{R}^{n \times K^2}$ .

Let  $L^{(k)}$  be a block diagonal matrix of column vectors  $\lambda^{(lk)}$  for  $l = 1, \dots, K$ .  $L^{(k)} = \text{diag}(\lambda^{(1k)}, \dots, \lambda^{(Kk)}) \in \mathbb{R}^{n \times K}$ .

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

Then  $P = XY^\top$ .

Similar to the  $K = 2$  case, we again have  $Y = X\Pi$  for a permutation matrix  $\Pi$ , so  $P = X\Pi X^\top$ . The permutation described by  $\Pi$  has  $K$  fixed points, which correspond to  $K$  eigenvalues equal to 1 with corresponding eigenvectors  $e_k$  where  $k = r(K+1) + 1$  for  $r = 0, \dots, K-1$ . 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  $(e_s - e_t)/\sqrt{2}$ .

So  $\Pi$  has  $K(K+1)/2$  eigenvalues equal to 1 and  $K(K-1)/2$  eigenvalues equal to  $-1$ .  $\Pi$  has the decomposed form  $\Pi = UI_{K(K+1)/2, K(K-1)/2}U^\top$ , and we can describe the PABM with  $K$  communities as a GRDPG with latent positions  $XU$  with signature  $(K(K+1)/2, K(K-1)/2)$ .

**Example** for  $K = 3$ . 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  $\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Another way to look at this is:

- Positions 1, 5, 9 are fixed
- The cycles of order 2 are
  - (2, 4)
  - (3, 7)
  - (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. Cluster membership identification (up to permutation).
2. Parameter estimation (estimating  $\lambda^{(kl)}$ 's).

Here, we will focus more on (1) and pose possible methods for (2). In our methods, we assume that  $K$ , the number of clusters, is known beforehand and does not require estimation.

#### 3.1 Previous work

Noroozi, Rimal, and Pensky [2] proposed using sparse subspace clustering (SSC) to identify the cluster memberships given either an edge probability matrix  $P$  or an adjacency matrix  $A$ . In the case that  $P$  is known, the cluster memberships can be identified exactly (up to permutation). A similar procedure can be applied if  $P$  is unknown and we have an observation  $A$ , but the theoretical guarantees of this method applied to the PABM are unknown. In particular, the method requires spherical Gaussian noise. The authors of this paper then use point estimators for  $\{\lambda^{(kl)}\}$  using cluster labels obtained via SSC.

#### 3.2 Clustering

##### 3.2.1 Using edge probability matrix $P$

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 clustering 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 [3].

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 clusters.

We can note from the explicit form of  $XU$  that it represents points in  $\mathbb{R}^{K^2}$  such that points within each cluster lie on  $K$ -dimensional subspaces. Furthermore, the subspaces are orthogonal to each other. Multiplication by  $Q \in \mathbb{O}(p, q)$  removes the orthogonality property but retains the property that each cluster is represented by a  $K$ -dimensional subspace. Using

this property, previous work proposes the use of subspace clustering while acknowledging some of its shortcomings [2] [5].

**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 of different clusters.

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

---

**Algorithm 1:** PABM clustering on the edge probability matrix.

---

**Data:** Edge probability matrix  $P$ , number of clusters  $K$

**Result:** Cluster assignments  $1, \dots, K$

- 1 Compute the spectral decomposition  $P = VDV^\top$ .
  - 2 Compute the inner product matrix  $B = |VV^\top|$ , applying  $|\cdot|$  entry-wise.
  - 3 Construct graph  $G$  using  $B$  as edge similarities.
  - 4 Identify the connected components of  $G$  and map each to cluster labels  $1, \dots, K$ .
- 

### 3.2.2 Using adjacency matrix $A$

The adjacency embedding of  $A$  approaches latent positions that form  $P$  as the number of vertices  $n$  increases. More precisely, let  $\{\lambda^{(kl)}\}_K \sim \mathcal{F}_K$  for some joint distribution consisting of  $K$  underlying clusters  $\mathcal{F}_K$ . Then the latent positions  $XU \sim \mathcal{G}_K$  for some related joint distribution with  $K$  underlying clusters  $\mathcal{G}_K$ . 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, Cape, Tang, and Priebe [3],

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

for some  $c > 0$  and sequence of  $Q_n \in \mathbb{O}(p, q)$ . In addition, Rubin-Delanchy et al. produce a central limit theorem result.

**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 clusters. Then for some  $c > 0$ ,

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

where  $\lim_{n \rightarrow 0} \rho_n = 0$  is a sparsity parameter introduced by Sengupta and Chen [4]. More precisely,  $\rho_n = o(\log n / \sqrt{n})$ .

In addition, by the central limit theorem,  $(\hat{v}_i^{(n)})^\top \hat{v}_j^{(n)}$  converge to a normal distribution centered at 0.

This leads to the following algorithm:

---

**Algorithm 2:** PABM clustering on the adjacency matrix.

---

**Data:** Adjacency matrix  $A$ , number of clusters  $K$

**Result:** Cluster assignments  $1, \dots, K$

- 1 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.
  - 2 Compute  $B = |VV^\top|$ , applying  $|\cdot|$  entry-wise.
  - 3 Construct graph  $G$  using  $B$  as its similarity matrix.
  - 4 Partition  $G$  into  $K$  disconnected components (e.g., using edge thresholding or spectral clustering).
  - 5 Map each partition to the cluster labels  $1, \dots, K$ .
- 

Theorem 4 implies that as  $n \rightarrow \infty$ , the number (not just proportion) of misclassified vertices, up to permutation, outputted by algorithm 2 goes to 0.

### 3.3 Parameter estimation

For any  $P$  edge probability matrix for the PABM such that the rows and columns are organized by cluster, the  $kl^{\text{th}}$  block is an outer product of two vectors, i.e.,  $P^{(kl)} = \lambda^{(kl)}(\lambda^{(lk)})^\top$ . Therefore, given  $P^{(kl)}$ ,  $\lambda$  and  $\lambda^{(lk)}$  are solvable exactly (up to multiplication by  $-1$ ) using singular value decomposition. More specifically, let  $P = \sigma^2 uv^\top$  be the singular value decomposition of  $P$ .  $u \in \mathbb{R}^{n_k}$  and  $v \in \mathbb{R}^{n_l}$  are vectors and  $\sigma^2 > 0$  is a scalar. Then  $\lambda^{(kl)} = \pm\sigma u$  and  $\lambda^{(lk)} = \pm\sigma v$ .

---

**Algorithm 3:** PABM parameter estimation using the edge probability matrix.

---

**Data:** Edge probability matrix  $P$ , cluster assignments  $1, \dots, K$ .

**Result:** PABM parameters  $\{\lambda^{(kl)}\}_K$

- 1 Arrange the rows and columns of  $P$  by cluster such that each  $P^{(kl)}$  block consists of edge probabilities between clusters  $k$  and  $l$ .
  - 2 **for**  $k, l = 1, \dots, K$ ,  $k \geq l$  **do**
  - 3     Compute  $P^{(kl)} = (\sigma^{(kl)})^2 u^{(kl)}(v^{(kl)})^\top$ , the SVD of the  $kl^{\text{th}}$  block.
  - 4     Assign  $\lambda^{(kl)} \leftarrow \pm\sigma^{(kl)} u^{(kl)}$  and  $\lambda^{(lk)} \leftarrow \pm\sigma^{(kl)} v^{(kl)}$ .
  - 5 **end**
-

A similar method can be applied using  $\hat{Z}$ , the adjacency spectral embedding of  $A$ .

---

**Algorithm 4:** PABM parameter estimation using the adjacency matrix.

---

**Data:** Adjacency matrix  $A$ , cluster assignments  $1, \dots, K$

**Result:** PABM parameter estimates  $\{\hat{\lambda}^{(kl)}\}_K$ .

- 1 Assign  $p \leftarrow K(K+1)/2$  and  $q \leftarrow K(K-1)/2$ .
  - 2 Compute  $\hat{Z}$ , the adjacency spectral embedding of  $A$  using signature  $(p, q)$ .
  - 3 Compute  $\hat{P} = \hat{Z}I_{pq}\hat{Z}^\top$ , the estimate of the edge probability matrix based on the adjacency spectral embedding.
  - 4 Arrange the rows and columns of  $\hat{P}$  by cluster such that each  $\hat{P}^{(kl)}$  block consists of estimated edge probabilities between clusters  $k$  and  $l$ .
  - 5 **for**  $k, l = 1, \dots, K$ ,  $k \leq l$  **do**
  - 6     Compute  $\hat{P}^{(kl)} = U\Sigma V^\top$ , the SVD of the  $kl^{th}$  block.
  - 7     Assign  $u^{(kl)}$  and  $v^{(kl)}$  as the first columns of  $U$  and  $V$ . Assign  $\sigma^2 \leftarrow \Sigma_{11}$ .
  - 8     Assign  $\hat{\lambda}^{(kl)} \leftarrow \pm\sigma^{(kl)}u^{(kl)}$  and  $\hat{\lambda}^{(lk)} \leftarrow \pm\sigma^{(kl)}v^{(kl)}$ .
  - 9 **end**
- 

**Theorem 5.** Under assumptions 1-6 as laid out by Sengupta and Chen [4], the estimators outputted by algorithm 4 converge to the true values as follows:

$$\sqrt{\frac{1}{nK} \sum_{k,l=1}^K \|\hat{\lambda}^{(kl)} - \lambda^{(kl)}\|^2} \xrightarrow{P} 0$$

*Proof. TODO*

The convergence rate of algorithm 4 is the same as that of the methods described by Sengupta and Chen [4].

## 4 Simulated Examples

### 4.1 Case $K = 2$

For these examples, we set the following parameters:

- Number of underlying clusters  $K = 2$
- Mixture parameter  $\alpha = .5$
- Within-cluster popularities  $\lambda^{(kk)} \stackrel{iid}{\sim} \text{Beta}(2, 1)$
- Between-cluster popularities  $\lambda^{(kl)} \stackrel{iid}{\sim} \text{Beta}(1, 2)$  for  $k \neq l$

#### 4.1.1 Clustering

In this part, we will assess Algorithm 2's performance for sample sizes  $n = 64, 128, 256, 512, 1024, 2048$ . For each sample size  $n$ , 100 sets of  $\lambda^{(kl)}$ 's are drawn and for each set of parameters, one adjacency matrix  $A$  is drawn and clustered. We will not



consider clustering using the edge probability matrix  $P$  since this will always result exact recovery of the original clusters (up to permutation).

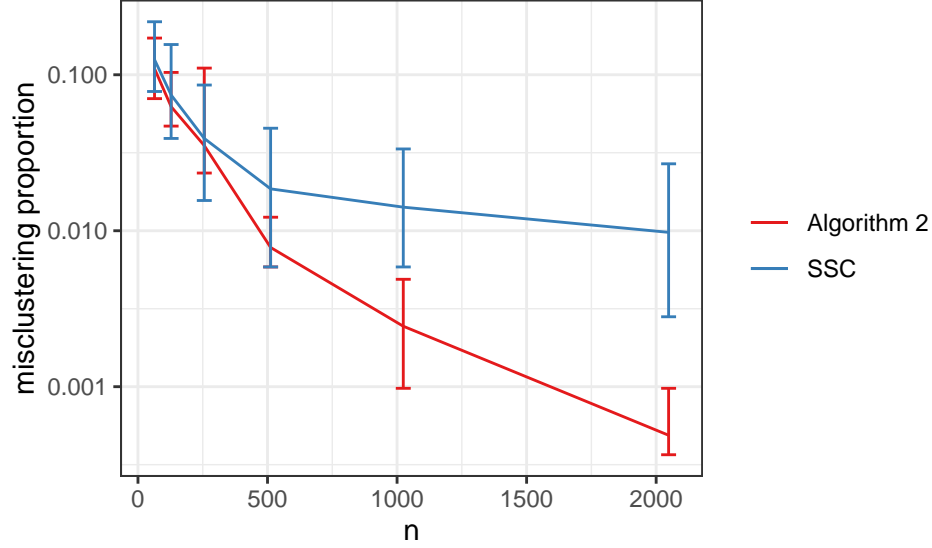


Figure 1: IQR clustering error using Algorithm 2 (red) compared against subspace clustering (blue) for sample sizes from 64 to 2048. Simulations were repeated 100 times for each sample size. The  $y$ -axis is shown on a logarithmic scale.

Theorem 4 implies that algorithm 2 will result in not just in the error rate converging to 0 but the error *count* as well (Fig. 2).

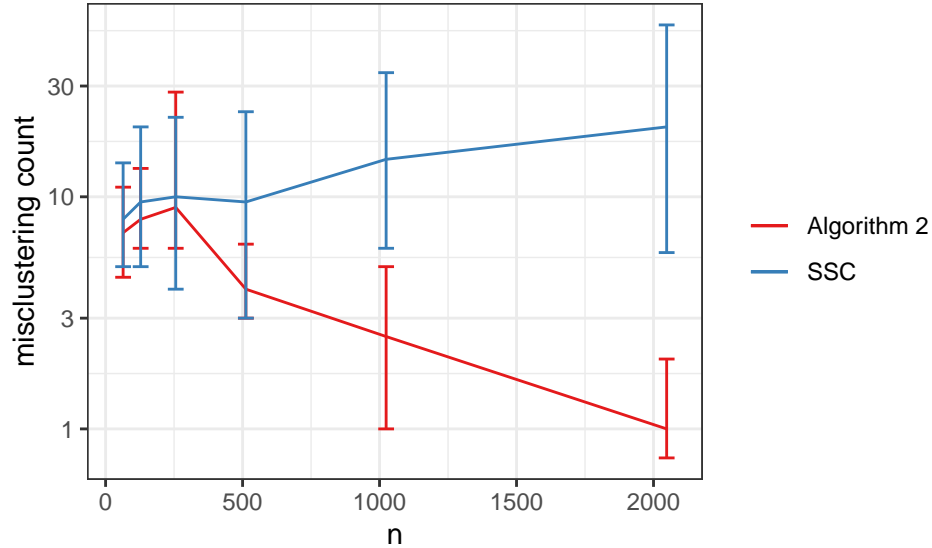


Figure 2: IQR of counts of misclustered vertices using Algorithm 2 (red) compared against subspace clustering (blue) for sample sizes from 64 to 2048. Simulations were repeated 100 times for each sample size. The  $y$ -axis is shown on a logarithmic scale.

We can also examine how the distribution of  $(\hat{v}_i)^\top \hat{v}_j$  varies with  $n$ :

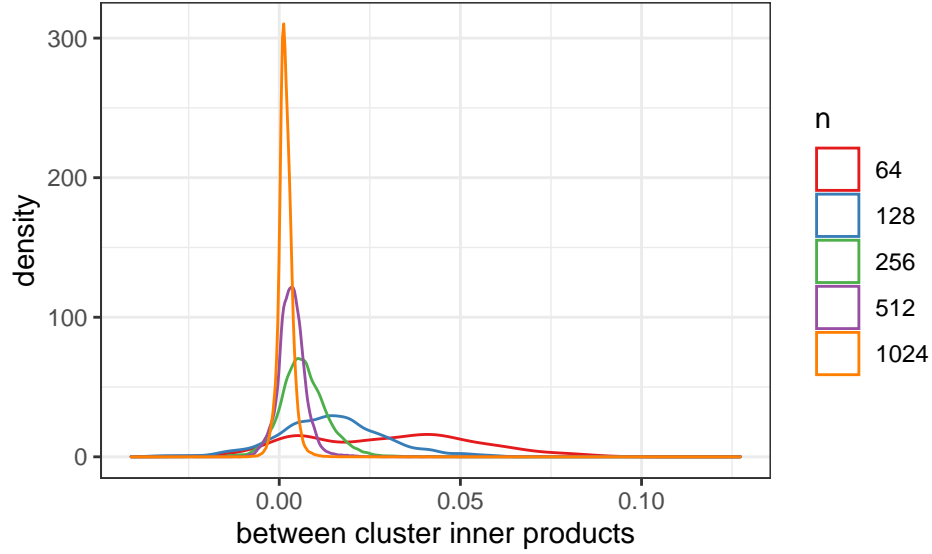


Figure 3: Between-cluster inner products of the eigenvectors of  $A$  for varying sample sizes.

#### 4.1.2 Parameter estimation

Figure 4 shows the medians and interquartile ranges of root mean square errors for Algorithm 4 over 100 simulations using the same parameters as before. Comparison against an MLE-based method [2] suggests similar performance, close to the rate  $n^{-1/2}$ .

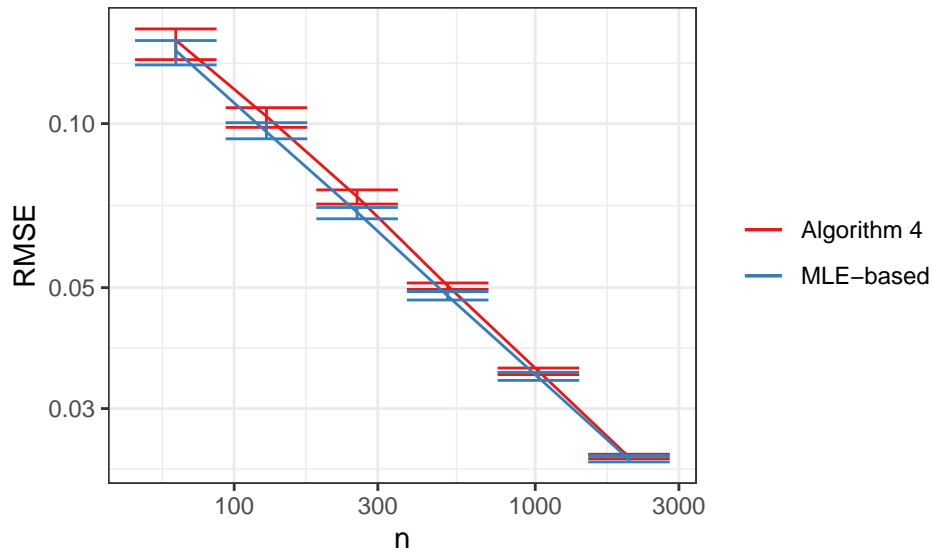


Figure 4: Log-log plot of the median and IQR RMSE from Algorithm 4 (red) compared against an MLE-based method (blue) using sample sizes from 64 to 2048. Simulations were repeated 100 times for each sample size.

## 4.2 Higher values of $K$

For these examples, we set the following parameters:

- Number of underlying clusters  $K = 2, 4, 8$
- Mixture parameters  $\alpha_k = 1/K$  for  $k = 1, \dots, K$
- Within-cluster popularities  $\lambda^{(kk)} \stackrel{iid}{\sim} \text{Beta}(2, 1)$
- Between-cluster popularities  $\lambda^{(kl)} \stackrel{iid}{\sim} \text{Beta}(1, 2)$  for  $k \neq l$

### 4.2.1 Clustering

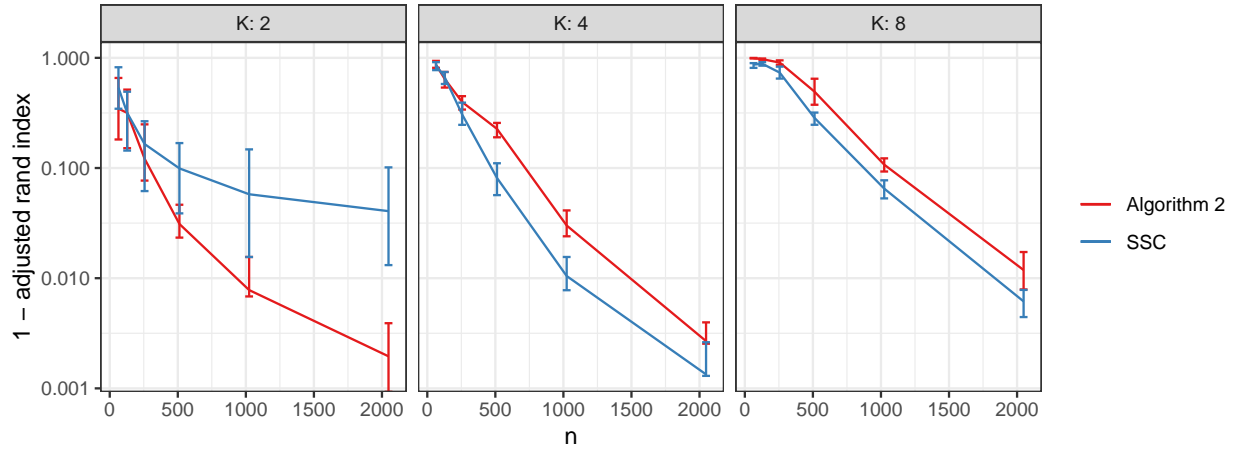


Figure 5: IQR of clustering error using Algorithm 2 (red) compared against subspace clustering (blue) for sample sizes from 64 to 2048 and number of clusters from 2 to 8. Simulations were repeated 100 times for each sample size. The  $y$ -axis is shown on a logarithmic scale.

#### 4.2.2 Parameter estimation

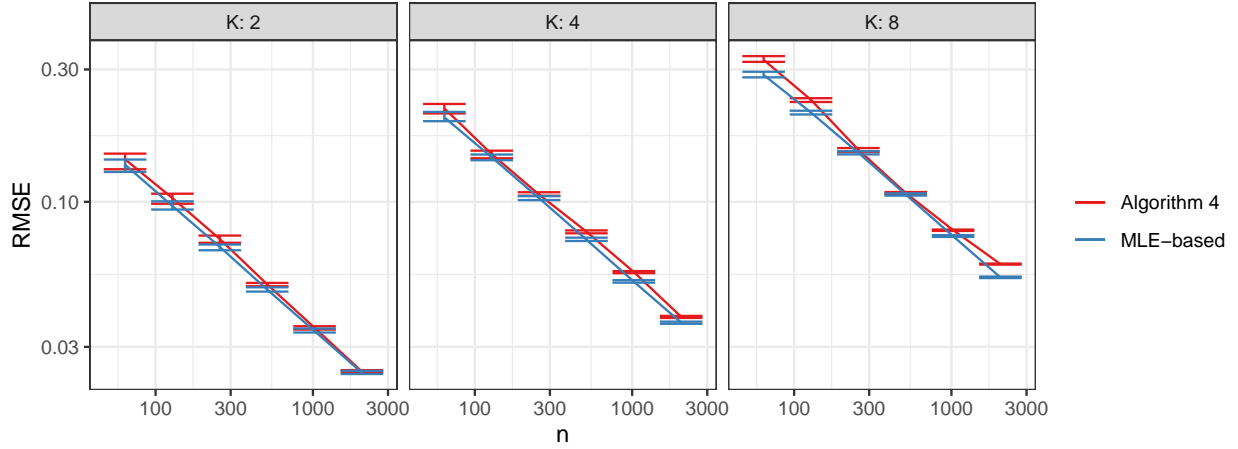


Figure 6: Log-log plot of the median and IQR RMSE from Algorithm 4 (red) compared against an MLE-based method (blue) using sample sizes from 64 to 2048 and number of clusters from 2 to 8. Simulations were repeated 100 times for each sample size.

## 5 Real data example

For the real data example, we applied Algorithm 2 to the Leeds Butterfly dataset [6] consisting of visual similarity measurements among 832 butterflies across 10 species. The graph was modified to match the example from Noroozi et al. [2]: Only the 4 most frequent species were considered, and the similarities were discretized to  $\{0, 1\}$  via thresholding. Fig. 7 shows a sorted adjacency matrix sorted by the resultant the clustering.

Comparing against the ground truth species labels, Algorithm 2 achieves an accuracy of 63% and an adjusted Rand index of 79%. In comparison, Noroozi et al. [2] achieved an adjusted Rand index of 73%.

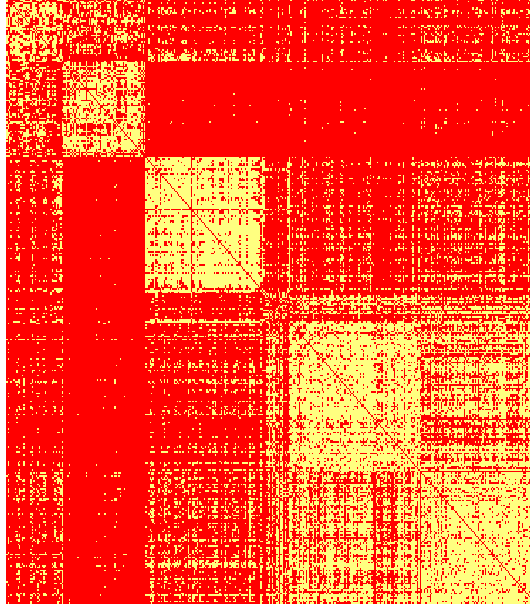


Figure 7: Adjacency matrix of the Leeds Butterfly dataset after sorting by the clustering outputted by Algorithm 2.

## 6 Discussion

### References

- [1] Avanti Athreya, Donniell E. Fishkind, Minh Tang, Carey E. Priebe, Youngser Park, Joshua T. Vogelstein, Keith Levin, Vince Lyzinski, Yichen Qin, and Daniel L Sussman. Statistical inference on random dot product graphs: a survey. *Journal of Machine Learning Research*, 18(226):1–92, 2018. URL <http://jmlr.org/papers/v18/17-448.html>.
- [2] Majid Noroozi, Ramchandra Rimal, and Marianna Pensky. Estimation and clustering in popularity adjusted stochastic block model, 2019.
- [3] Patrick Rubin-Delanchy, Joshua Cape, Minh Tang, and Carey E. Priebe. A statistical interpretation of spectral embedding: the generalised random dot product graph, 2017.
- [4] 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.
- [5] 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>.
- [6] 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>.